University of Groningen

Software Engineering

Evidencio Architecture Document

Authors:
Aleksandar Sasa Janjanin
(s3169618)
Jaap van der Vis
(s2344076)
Tomasz Kuczak
(s3619109)
Javier Png *(s3611655)*
Siheon Lee *(s2898373)*
Gizem Aydin *(s3611523)*
Dammes de Zoeten
(s2892138)

Lecturer:
Mircea Lungu
Teaching Assistant:
Frank te Nijenhuis

April 17, 2018

1 Introduction

Evidencio Patient Platform is a website where patients can use Evidencio prediction models to gain insight in medical decisions and their consequences. The platform has a tool to enable the medical professionals to create their own workflows by importing the Evidencio models and modifying them in a patient friendly way. Thus, the patients can fill in the workflows with their data easily. Moreover, the Portal presents the prediction results to the patients using understandable graphics, and the patients are able to save these results so that they can consult to their doctor.

2 General overview

The platform is a website that consists of two components:

- The patient side, where users can follow the workflows and eventually receive model calculation results.
- The designer side, where medical professionals can create and edit workflows.

3 Patient

3.1 Searching for Workflow

Users will search for workflow models using a standard search bar. The search will work by comparing the input string to keywords attached to the workflow models created and saved in a database. As of now we do not have any saved workflow models. Therefore, we are passing the search query to Evidencio's REST API that returns all the models from the Evidencio website matching the search criteria.

3.2 Complete Workflow Model

Once the user has chosen the workflow model they would like to complete. A SQL query will be sent to the database asking for the the variables of the first step of the workflow. Variables have two input types, catagorial and continous. When the user has filled in the inputs for the variables, the webpage will either load a new step or route to the result page. If the webpage needs to load a new step, an API call will be sent to Evidencio's

REST API and the result will be sent through our rule engine, which will produce the next correct step for that workflow.

As of now we do not have any saved workflow models. Therefore, we are only working with the Models from the Evidencio website. All the required input variables for the choosen Workflow Model are recieved through an API call to Evidencio's REST API. After the user submits the required values for the variables, a http POST request is sent to the result page.

3.3 Graph of Workflow Model Result

After the Worklow model is completed the result of the workflow will be presented in a new page. The result is calculated by taking the queries that were sent through http POST request, and then use them for a API call to Evidencio's REST API that will calculate the result. Using the result the page will create graphical representation using Chartjs. The user is able to change the charts to a pie chart, bar chart, doughnut chart and a polar area chart. If there is any additional result text, it will show that below the graph.

3.4 Exporting of Results to PDF

The patient can choose to export the results in a PDF format. Since the results will not be stored in the server, the patient can save the results as well. This will provide the doctor with a more appropriate results for the doctor to interpret and make better decisions

The creating of PDF will be done using TCPDF, a library that is commonly used by websites worldwide. A HTML structure will be predefined, and the results will be retrieved using session and put in to the HTML wherever applicable.

4 Designer

4.1 Design of Workflow

Medical professionals can make workflows based on Evidencio-models at the designer side of the website. A workflow consists of Steps, which can either be Input-steps or Result-steps. The variables, which are loaded via an API-call from `Evidencio`, can be placed in steps, and their representation can be changed. Currently only the title and description can be changed, in the

future it will be possible to also change the widget type used (slider, text, radio, etc.).

The designer can also configure the API-call done to `Evidencio` at the end of a step. He can select the model that he wants to calculate and choose the variables used for the API-call. This information can be used to make API-calls once the user has filled in the variables. Since variables are not automatically selected, the designer can reuse previously filled in variables (in case of multiple models with the same variables).

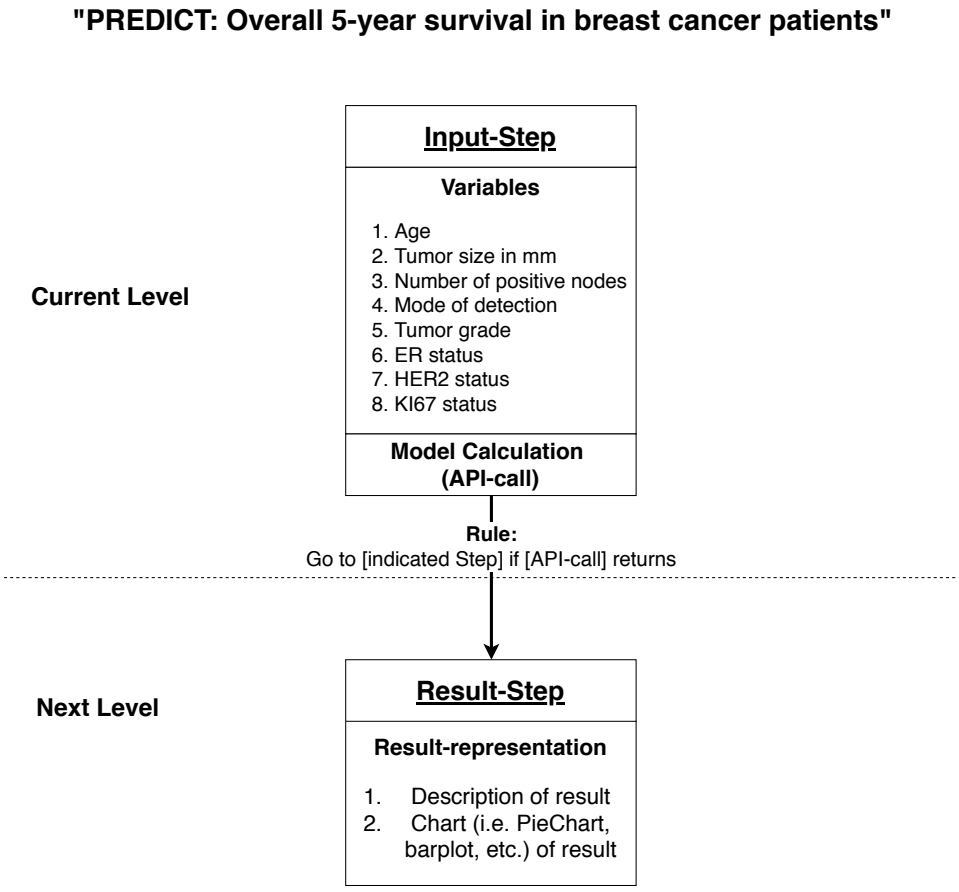

Figure 1: Workflow with one Input-step and one Result-step. The relation between the Steps is based on a Rule. This rule points to the Result-step as soon as the Model calculation (API-call) is finished.

The JSON-rules-engine[1] is used for the logic between Steps. A Rule in the engine consists of conditions and an event. The event is executed if the conditions are true. Possible uses are, for example:

1. The designer wants to go from the Input-step to the Result-step as soon as the API-call returns a result. A rule can be used to go to the Result-step, depending on if the API-call is done or not. See figure 1 for an example.

2. The designer wants to show a different graph/description based on what the result is. Using rules the designer can specify which Result-step (each step with their own graph/description) should be shown, based on the calculated result. See figure 2 for an example.

4.2 Verification of Workflow

When the Workflow is completed, the designer will be submit the workflow for verification. A certified user will then receive the request for verification and can verify the workflow. This will be included to ensure the quality of models and to prevent possible (design-related) mistakes. After verification the workflow will be available for a patient to find.

4.3 Sidebar

There is a dynamic sidebar that is used to navigate through the pages of the designer side. The designer can access this sidebar from every other page when he is logged in.

4.3.1 My Workflows

The aim of this page is to enable the designer to manage his workflows. The designer can add a new workflow and he can delete/edit his existing workflows. It is also possible to filter the workflows by their status. After establishing a database, this page will get information about designers' workflows from the database, and modify the database according to the designers' actions.

[1] JSON: rules engine - `https://www.npmjs.com/package/json-rules-engine`

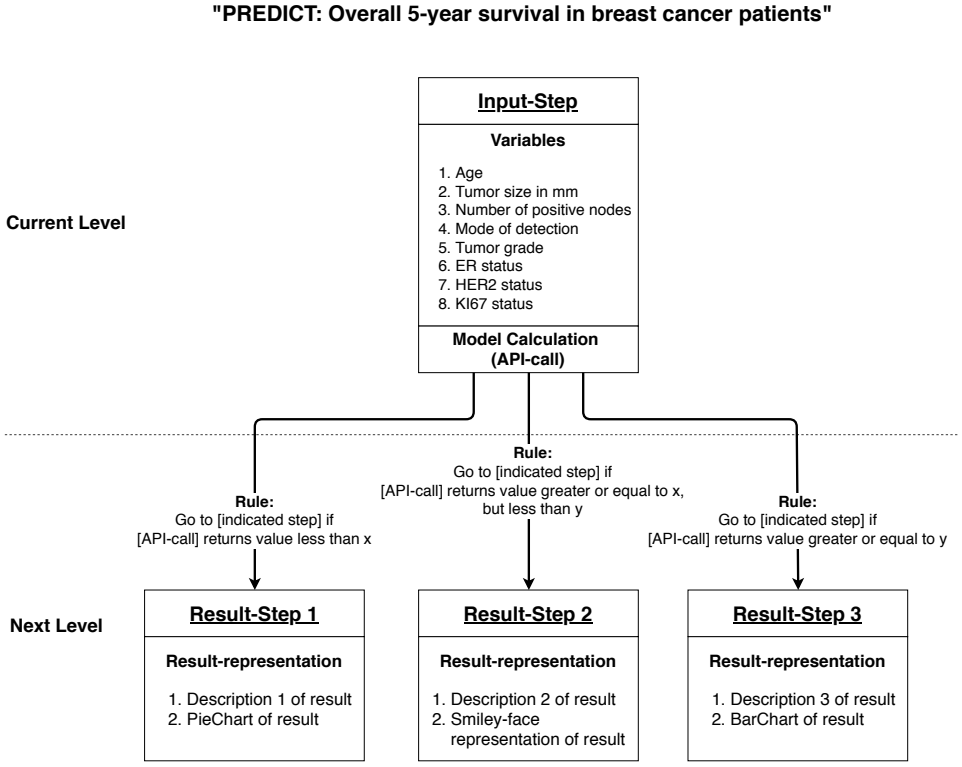

Figure 2: Workflow with one Input-step and multiple Result-steps. The relations between the Steps are based on rules. Based on the outcome of the Model calculation (API-call) the Result-step is chosen. Each Result-step has its own description and representation type.

4.3.2 Administrator

This page will only be available to the administrator designers. Administrators are certified users who has the authority to verify workflows.They will be able to verify workflows and answer the questions of other designers using this page. This page is still under construction and he details of implementation have not been determined yet.

4.3.3 Edit Account Details

The designer will be able to view and edit the details of his account using this page. This page is still under construction and he details of implementation have not been determined yet.

4.3.4 Instructions

This page will contain information about using the designer side. The designer will be able to learn about creating workflows, sending them for verification, etc. This page will be implemented once the designer side is finished.

4.3.5 Contact Us

This page will allow designers to submit their questions related to the webpage so that administrators can answer them. This page is still under construction and he details of implementation have not been determined yet.

5 Other

5.1 Authentication and authorisation

Authentication is required only to access the designer side and is based on standard Laravel authentication. It is expanded, so that during registration, the medical professional provides his data and documents that prove his medical status. Then his account needs to be verified by an administrator.

There is a permission that can be granted to medical professionals that allows them to review and accept new workflows.

5.2 Communication with the Evidencio API

The platform uses Evidencio REST API in the designer side to map input variables of the models to the ones available in the designed workflows. In the patient side, the API is used to calculate a model's result after completing a workflow. Since the API needs a key to work, it needs to be provided in the configuration file before running the website.

5.3 Database

The database was designed following the Object-Role Modeling syntax. However, the diagrams can be easily converted to plain SQL tables. The design consists of two parts - first is focused on entities related to workflows (Figure 3), and second covers Users, reviewing workflows, and all other entities (Figures 4 and 5).

6 Team organisation

The team is divided into three subteams:

- "Patient" (2 people) - responsible for the patient side.
- "Designer" (3 people) - responsible for the designer side.
- "Rest" (2 people) - responsible for other aspects such as the project setup, database, authentication, authorisation and Evindencio API.

  "Patient" and "Designer" teams are responsible for both backend and frontend of their components.

7 Technology stack

The backend is written in PHP, using the Laravel framework. MySQL is used for the database system. We use Laravel's Eloquent ORM for communication with the database.

The frontend uses Bootstrap and Vue2.js. The Vue-Multiselect[2] component of Vue2.js is used for the select-input used for the variables.

The workflow designer uses the Cytoscape JS library[3], which is a purely Javascript library made for graph theory. The Cytoscape-Canvas add-ons is used for drawing on the background of the graph. Communication with Evidencio API is achieved using Guzzle.

The JSON: rules engine[4]

8 Changelog

[2] *Vue-Multiselect*, Damian Dulisz, `https://vue-multiselect.js.org`

[3] *Cytoscape.js: a graph theory library for visualisation and analysis*, Franz M, Lopes CT, Huck G, Dong Y, Sumer O, Bader GD, `js.cytoscape.org/`

[4] *JSON: rules engine*, Cache Hamm, `https://www.npmjs.com/package/json-rules-engine`

Contributors	Date	Section	What Was Done
Tomasz	27-03-2018	All	Added initial document layout and 'Other' section
Dammes	27-03-2018	Patient	Added patient side
Jaap	28-03-2018	Designer, Technology	Added Designer architecture and Technology
Gizem	16-04-2018	Introduction	Modified Introduction
Gizem	16-04-2018	Designer	Added Sidebar
Jaap	16-04-2018	Designer	Updated Designer/Technology
Dammes	16-04-2018	Patient	Updated patient side
Tomasz	17-04-2018	Other	Added database description and diagram

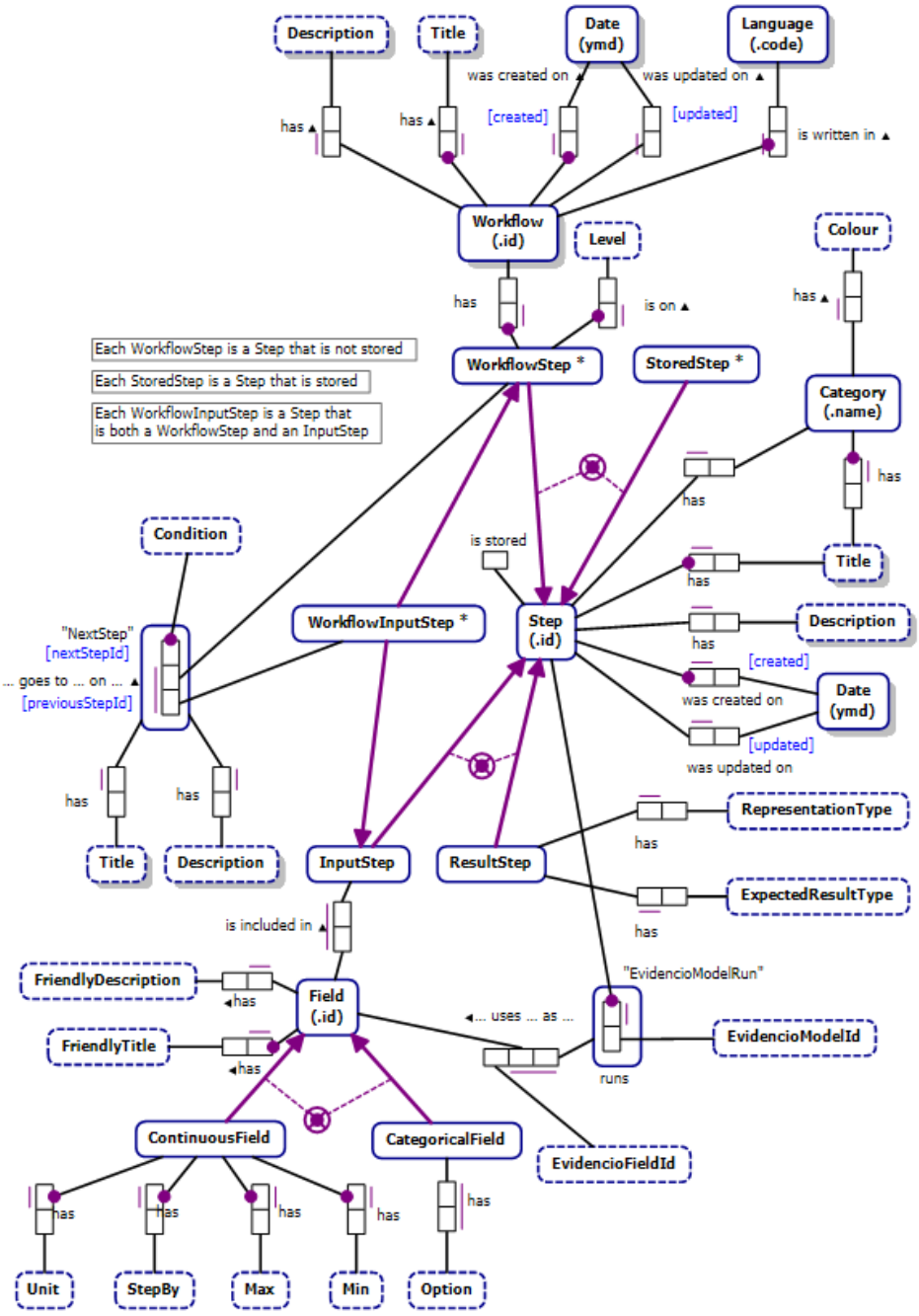

Figure 3: ORM diagram of workflow-related entities.

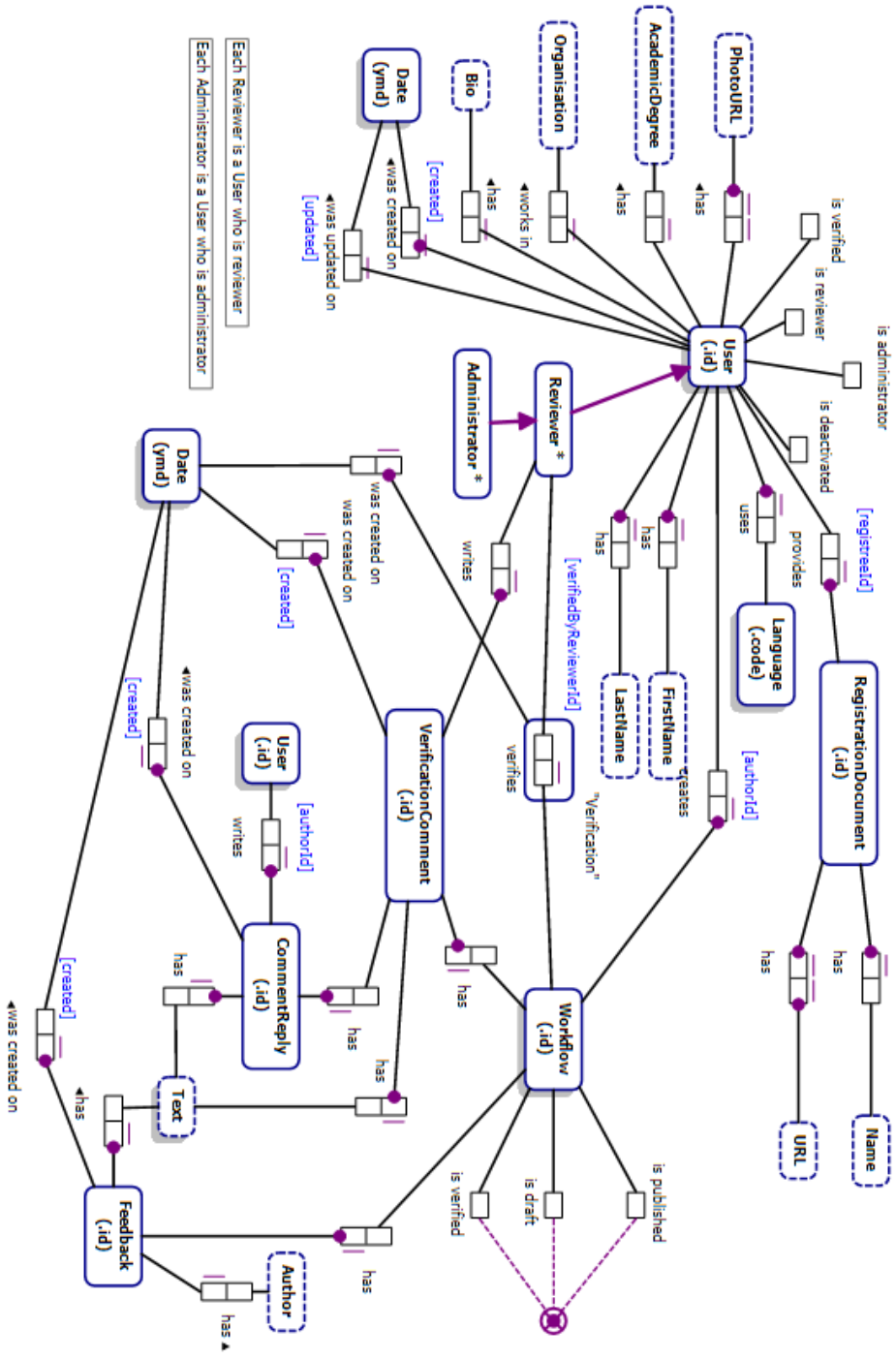

Figure 4: ORM diagram of User and review-realted entities.

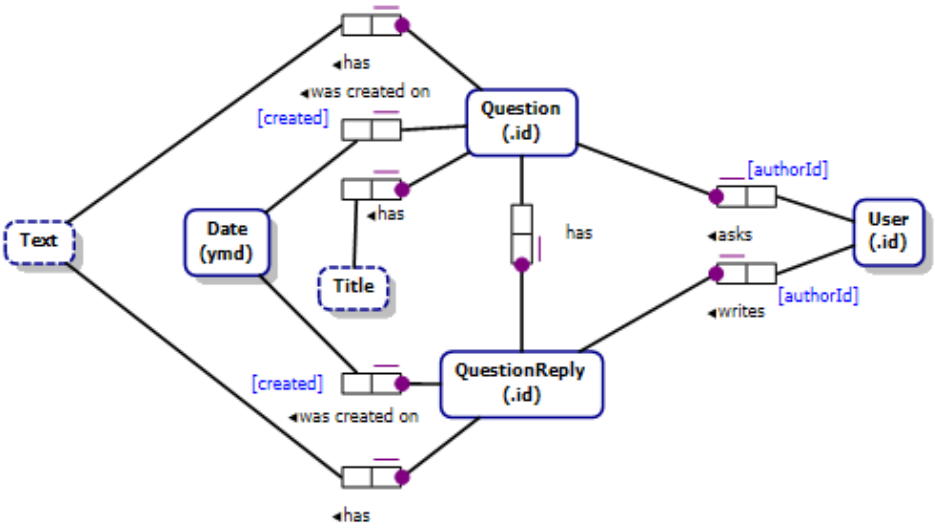

Figure 5: ORM diagram of Question and QuestionReply entities.