MELANOQ

Valentina Cecchini - m. 255596
Stefano Valentini - m. 254825

Bio Informatics, University of L'Aquila, July 2019.

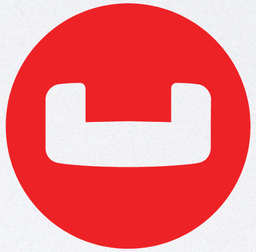
TECHNOLOGIES

- To offer the best decoupling and deployment flexibility we decided to follow the classic **MVC** web-application architectural pattern by developing a **REST Server** and a **Web Client**, while storing the data on a **NoSQL documental database**.

- The execution flow follows the MVC paradigm: the data stored in the database is presented to the Web Client, from it it is possible to invoke functions on the REST Server that will then modify the aforementioned data, updating the Web Client view.

DATABASE

- Is a **documental NoSQL database** whose data structure is in **json** format, that is a simple, lightweight notation, compact and human readable.
- Usually a document is a single instance of an object in the application code and can contain nested structures, that allows developers to naturally express many-to-many relationships.
- Offers the possibility **distribute** the storage.
- **N1QL** (pronounced "nickel") is Couchbase's query language and aims to meet the query needs of distributed document-oriented databases. It is very similar to SQL.



Couchbase

REST SERVER

- **Spring Boot:** makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run".
- **Spring Data JPA for CouchBase:** Spring Data for Couchbase is part of the umbrella Spring Data project which aims to provide a familiar and consistent Spring-based programming model for new datastores while retaining store-specific features and capabilities.
- **Spring Security:** Spring Security is a powerful and highly customizable authentication and access-control framework.
- **Json Web Token-based authentication:** is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object.

WEB CLIENT

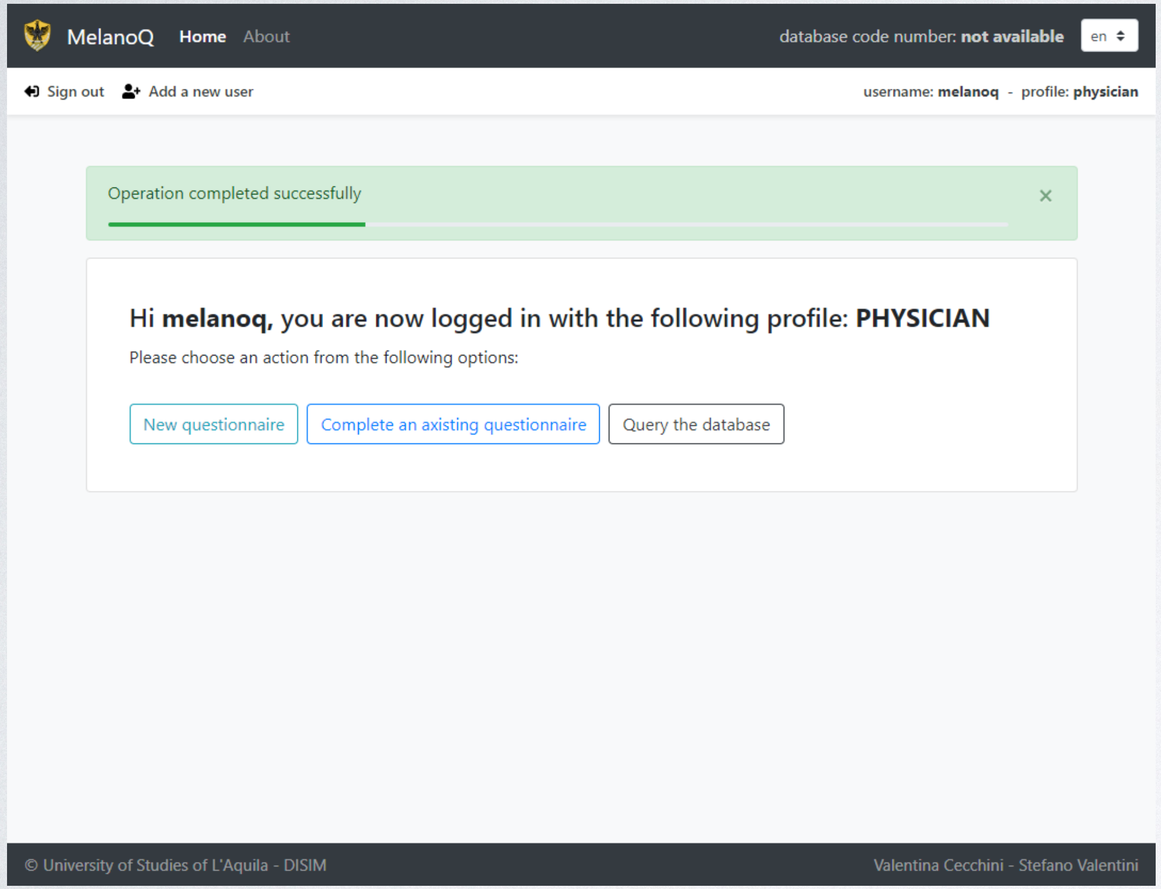
- **VueJS**: it is a progressive framework for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries.

DATA MODEL

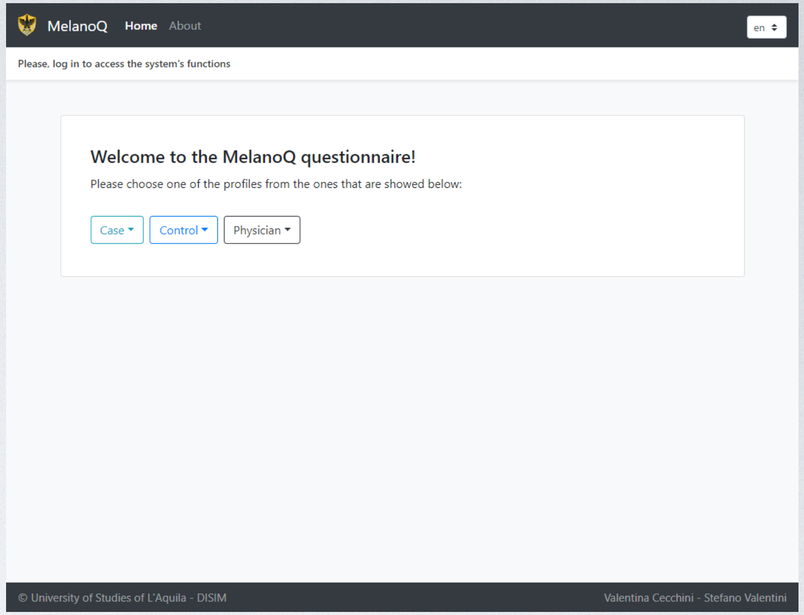
```
{
    ...

    "a2": {
        "education": "text",
        "ethnicity": "text",
        "currentOccupationalStatus": "text",
        "sex": "text",
        "weight": "numeric",
        "historyOfOccupations": {
            "occupationStartingTime": "text",
            "occupationEndingTime": "text"
        },
        "provinceOfBirth": "text",
        "countryOfBirth": "text",
        "residencies": {
            "residencyEndingTime": "text",
            "residencyStartingTime": "text",
            "residencyCountry": "text",
            "residencyCity": "text"
        },
        "height": "numeric"
    },
    ...
}
```

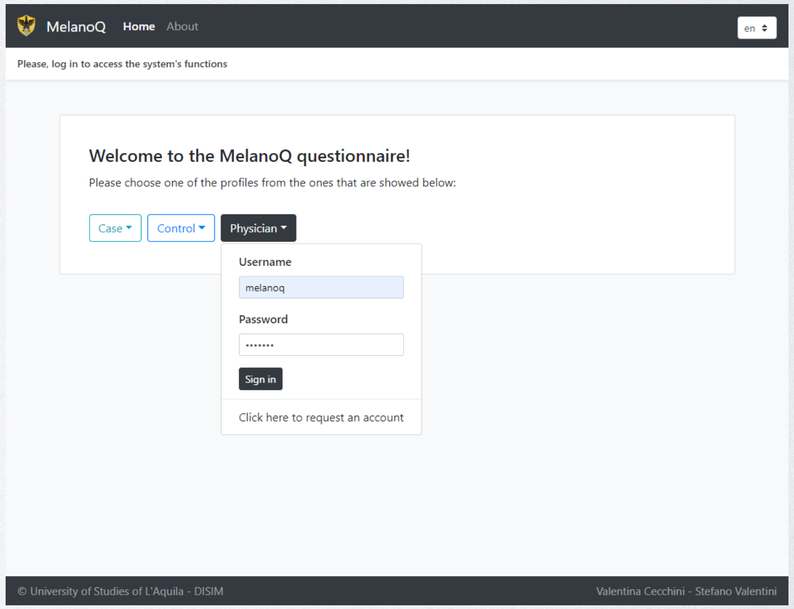
CLIENT HOME



Home view as logged user

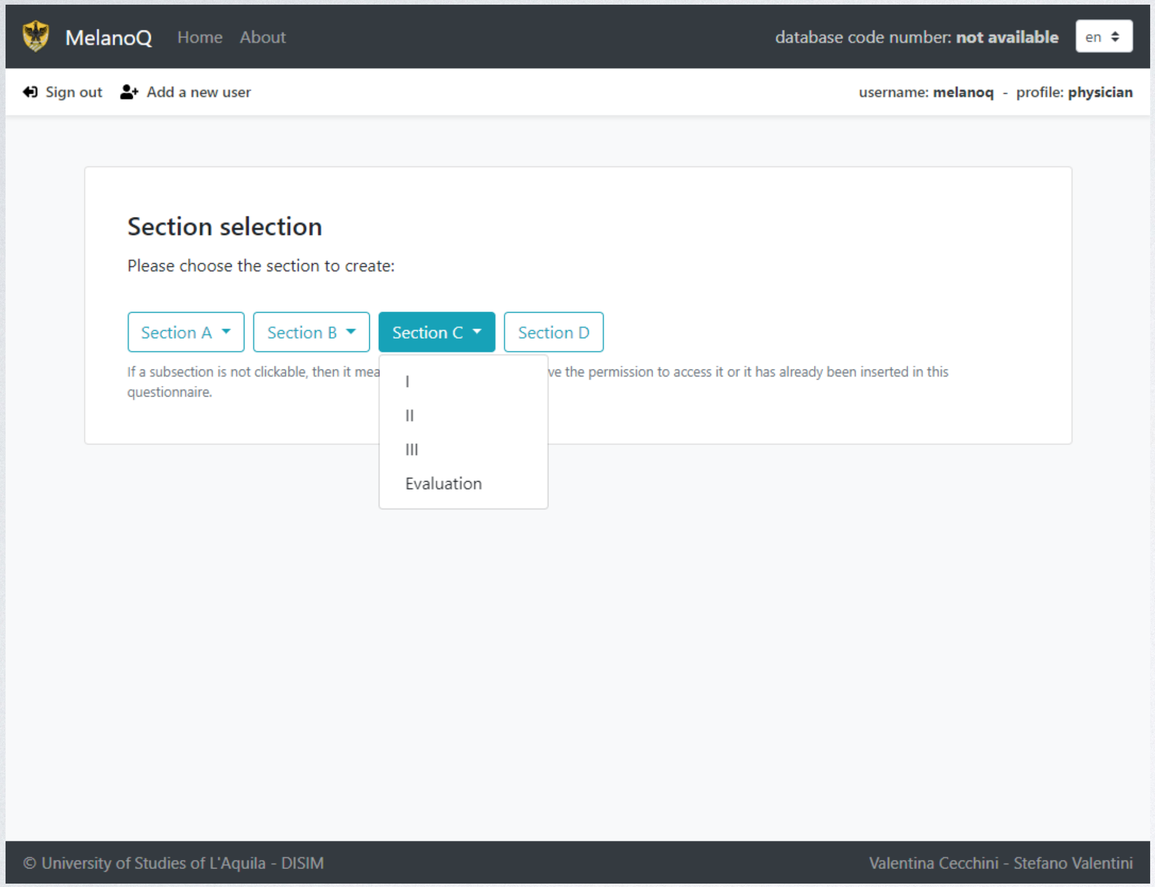


Home view as guest

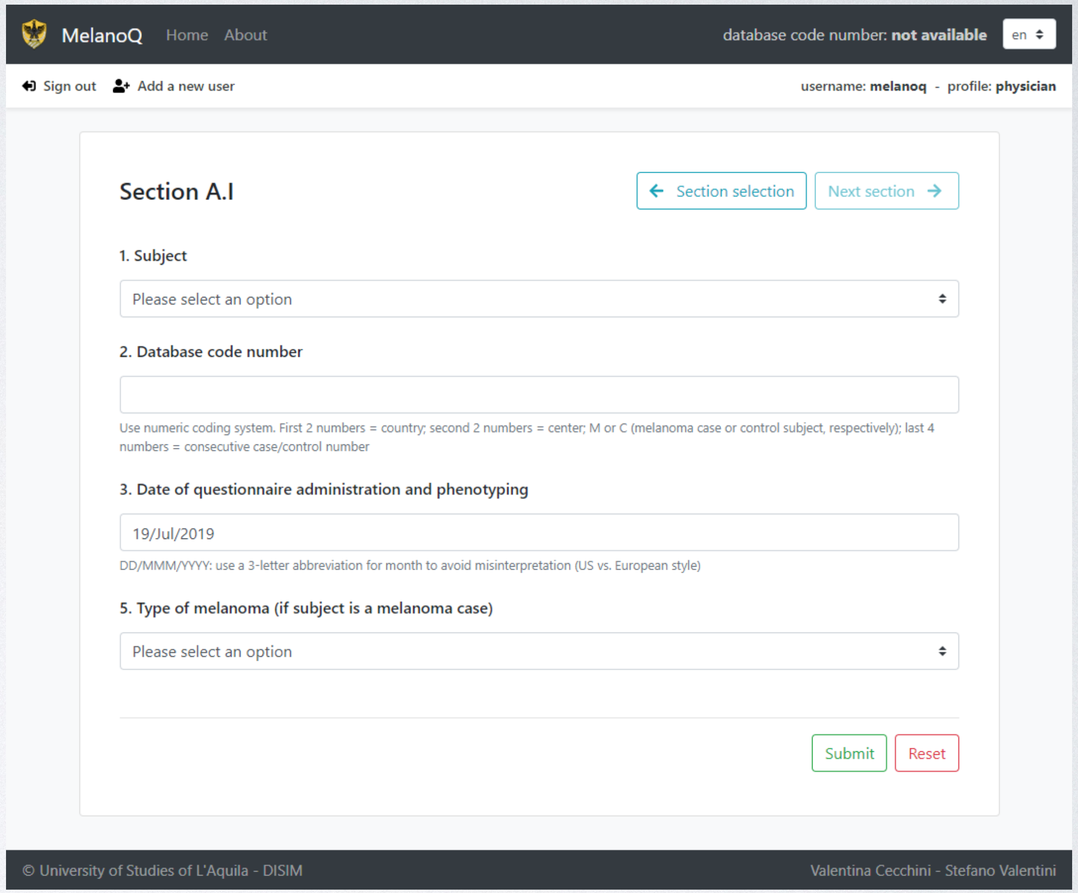


Home view: log in

SESSION SUBMISSION

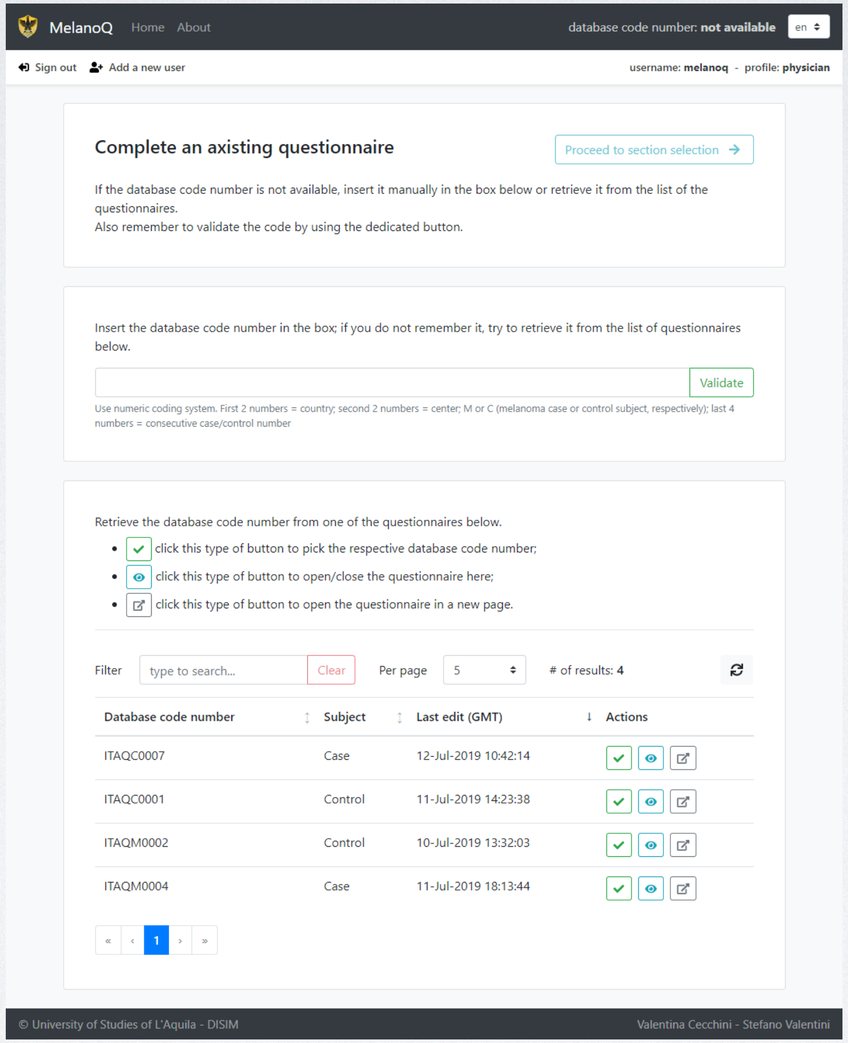


Section selection

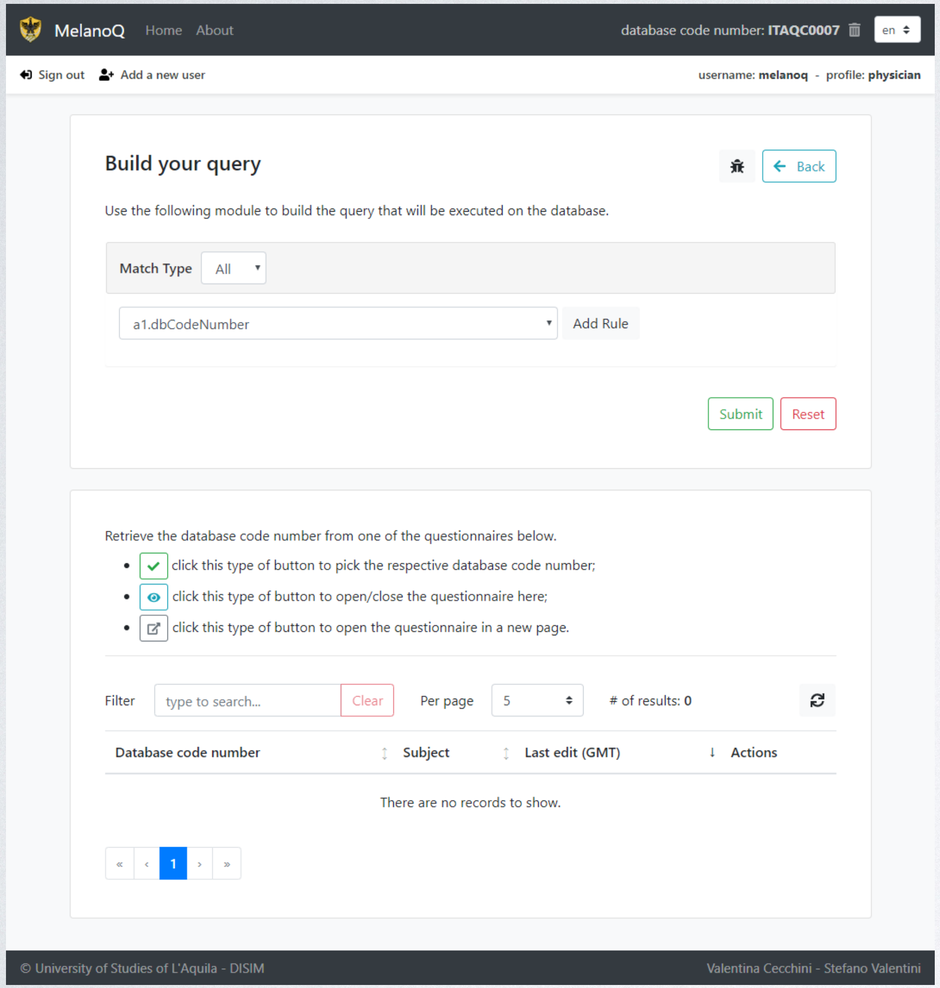


Section from example

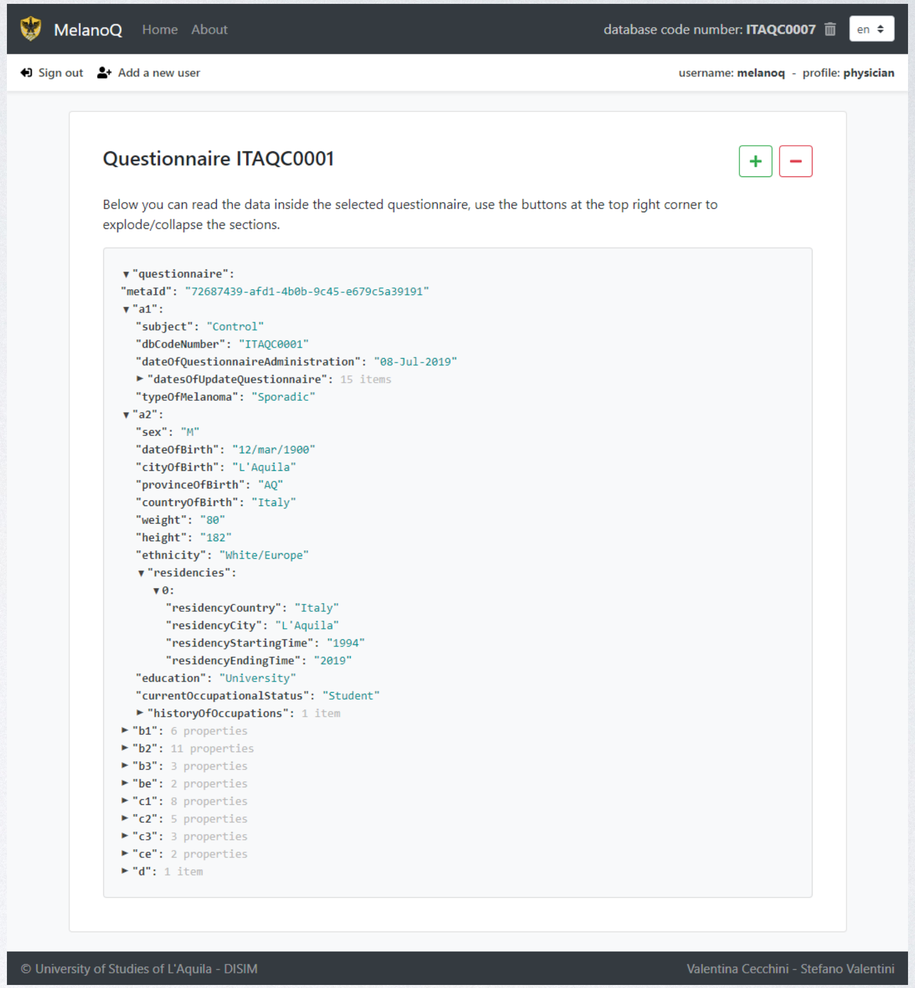
DATABASE CODE NUMBER RETRIEVAL



QUERY BUILDER



Query builder view



Standalone view

Thanks for the attention.