Case study: tracking and managing human rights complaints

Summary

The present case study is based on the development of an information system to handle victim's complaints at a human rights organization. Complaint reception and management is a recurring task for national and international institutions in the human rights field.

Given the fact that most of the time this information system feeds legal procedures (either locally or internationally), at some point, the data structure should reflect the legal complaint path, to track a specific case effectively. Having said that, it's worth remembering that human rights complaints management does not always have a later development in the judicial arena, and it instead reflects internal managerial tasks (or it gets mixed with other data purposes as well). This guide will attend both documentation scenarios.

This study is based on a real project, but does not include any information regarding a particular institution or country to protect our partners' privacy. Furthermore, it will consider alternative scenarios or experiences for didactic purposes. This way, you will gain a more comprehensive view on complaint management as it relates to human rights institutions.

This case study describes how an organization carried out the following steps in building an information system:

- Identify information system needs.
- Describe the complaint management workflow.
- Convert the information management needs into technical requirements for the information system.
- Determine how the data structure and the properties we use to describe entities (the objects described in the database) are connected with the information needs.
- Ensure that data is linked to specific outcomes that the system is expected to provide.

This document is organized by first describing the project objectives and information needs. Then, it presents the discovery phase and explains in depth how the database was designed to address the project requirements, to end with learnings and recommendations for future projects in which potential junction points are identified.

Case study specification

Objectives of the project

The general goal of the project under study was to develop and implement an information system to receive, track and manage complaints related to human rights issues.

Complaints are presented by victims or complainants (they could be the same or different persons) using a submission form on a website. This step launches the handling process. Once the admissibility of the complaint is approved, it passess through several stages until it is closed. In this example, the case is closed for the purpose of this organization when the complaint reaches an endpoint in the workflow.

As we will see in the workflow later in this document, the case can be closed in several ways: the complaint is not admitted, the complaint has no substance, a conciliation agreement has been signed and verified, the complaint has been referred to another authority, or legal proceedings are instituted, meaning that the case is referred to to a higher authority.

This information system was implemented using Uwazi, the free open-source software developed by HURIDOCS for organising, analysing and publishing information. For this project, as we will see later in the database design section, the platform was adapted to handle complaints and provide valuable insights about the internal performance of the partner.

The discovery phase: from an idea to a data model design

The purpose of the discovery phase is to gain understanding about the partner' needs and the challenges they want to overcome from their preexisting data model and/or workflow. Therefore, the design of the new database took into account these considerations:

- Partner' needs and challenges they wanted to overcome.
- What an actual complaint procedure looks like and the different responses it could get: this required research on existing complaints proceedings, and designing

potential scenarios. This resulted in a schema of how information would travel in the data system.

- Design of templates (the forms with a set of fields that are used to describe an entity) and data flow.

- The internal dynamics of the organization, including:

    - Levels of information access within the organization (access policy).
    - Distribution of tasks and responsibilities: who is responsible for adding or updating information at each stage of the workflow, which starts with the raw complaint, as it is presented by the complainant.

To surface this valuable information, various **discovery techniques** were carried out in several digital and face-to-face meetings:

1. Database user stories: listening to short descriptions of software uses and requirements from different user profiles is essential to reach an effective solution. These user profiles should reflect the variety of tasks performed in the database, the levels of tech savviness in the organisation, the access permissions, the degrees of involvement in the database, etc. This project involved very different profiles: managerial, investigators, communications practitioners, etc.
2. Feature identification: two inception meetings were carried out to identify the mandatory and desirable features of the database. Participants identified relevant features for their work individually and then assessed the features identified by the rest of the team.
3. Elevator pitch: this exercise helped to build a concise statement describing the target groups, their needs, the definition of the tool and its main functionalities. These were some of the findings:
    a. Target groups: victim, complainant, respondent (the alleged perpetrator), lawyers, organization's personnel, etc.
    b. Needs: documentation, tracking, statistics, reporting, confidentiality, internal complaint handling.
    c. Tool functionalities: collecting, tracking, research, analysis, labelling, reporting (statistical outputs).

4. Research: additional investigation on complaining tracking was done to identify how the process starts and how it can end (potential scenarios).

As a result, we identified the main challenges this organization wanted to overcome and the functionalities of the new information system to address these challenges.

Challenges	Information system functionalities
To handle complaints more efficiently by reducing time for repetitive tasks.	Automatically generate notifications for victims, complainants and respondents at certain stages of the project. Register complaints directly from a submission form, which is the main source of information for the database. Handle complaints through several stages, updating or adding information at every stage.
To get a quick overview of all the information regarding a particular complaint.	Reflect the evolution of a complaint in a timeline, highlighting different stages. Record information in several formats: text, video, audio and photo.
To get an overview about the characteristics and the status of complaints as a whole (total of active, closed or pending complaints, number of complaints at each stage, etc.).	Retrieve and count complaints by parameters such as date, place, status, staff member who attends the complaint, characteristics of the complainant, characteristics of the respondent, etc.
To extract relevant indicators about complaints as a whole in a given period of time such as types of violations, geographical spread,	Extract aggregative data statistics, reports or graphics about multiple cases based on common relevant parameters.

characteristics of victims (gender, age, occupation, vulnerable group) and respondents, etc.	
To monitor the performance of the organization to evidence if complaint management goals are fulfilled.	Reflect the efficiency of complaint management by the team with indicators such as the time and the personnel involved per case.

The design of the database

Once the main requirements of the project were identified and prioritized with the team, we moved forward with designing the database structure. The needs identified at the discovery phase guided this process.

In this case, the data model should reflect the different stages of the complaint handling process. As we will show later, each stage takes information from the previous one and updates or completes it with new information. For example, the stages for this process included:

0. *Complaint submission:* the information of the original complaint comes straight from the complainant.
1. *Complaint registration*: the complaint is registered by the personnel of the organization who is in charge of categorizing it and assigning it to a person responsible for handling the case.
2. *Initial investigation*: If the complaint is admitted and consequently investigated, new information regarding the investigation is added, and so on.

However, this project entailed particular information needs about the internal work of the organization and thus, the model was customized to add functionalities to monitor the organizational performance. These functionalities resulted in the dashboards that we present in the analysis section. The workflow stages constitute the main part of the data structure, so it is very useful to see how the process looks graphically (the numbers on the left side correspond to the stages mentioned in the previous example).

Complaint workflow

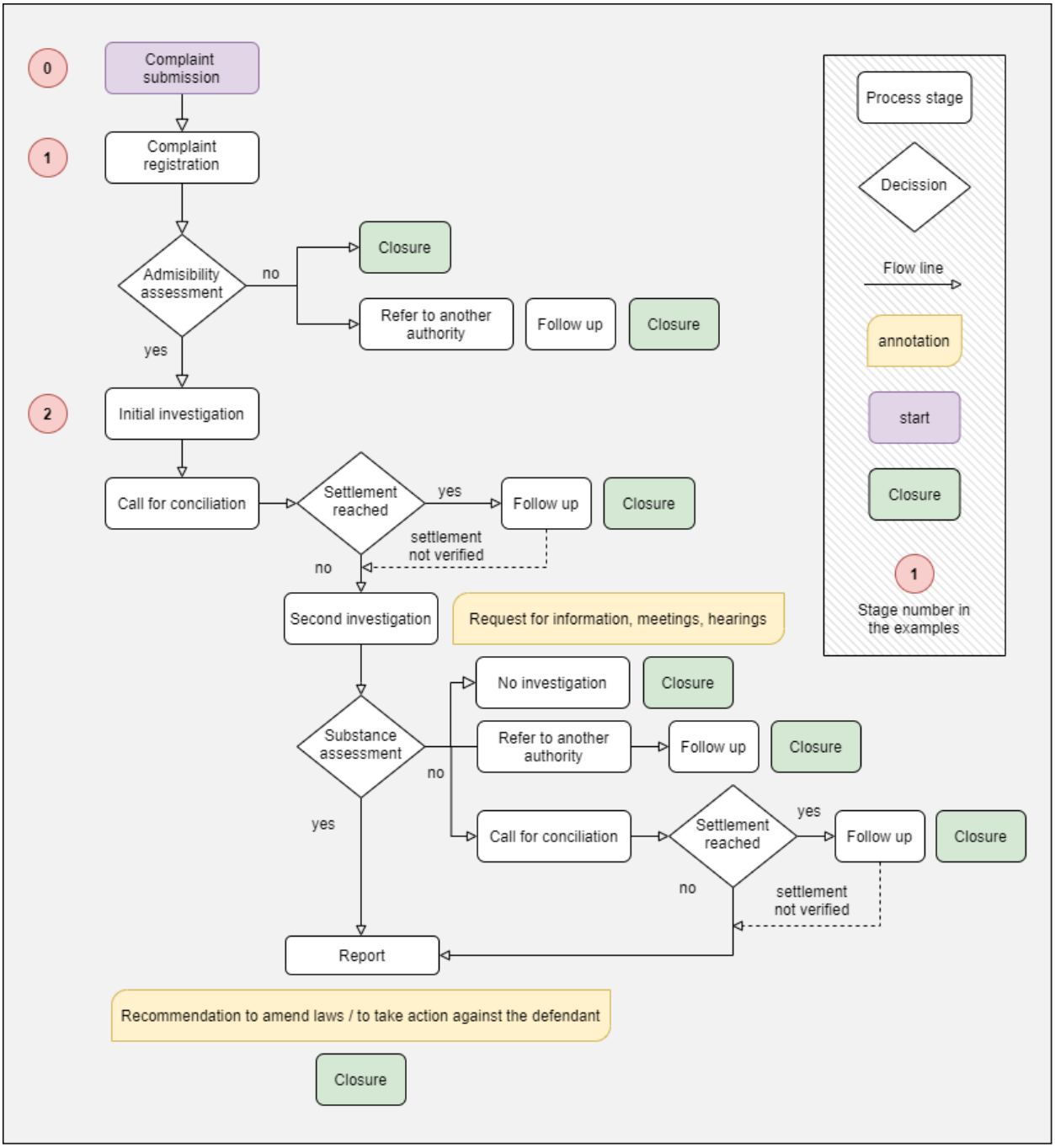


Diagram 1. Complaint workflow

Data model

The insights that were identified through the discovery phase and with the help of the data flow for this project were translated into the queries that the database is able to answer; then we listed the information needed to respond them and draft how to organize it, according to information buckets (types of entities described in the database, called *templates* in Uwazi), establishing relationships between them, and defining the possible description values (called *thesauri* in Uwazi).

Almost *every process stage* in the workflow constitutes a new entity in the database: *complaint submission, complaint registration, initial investigation, second investigation, call for conciliation, closure*, etc. However, all these elements, except *complaint submission* and *complaint registration*, are gathered within the same type of element called *complaint stage*:

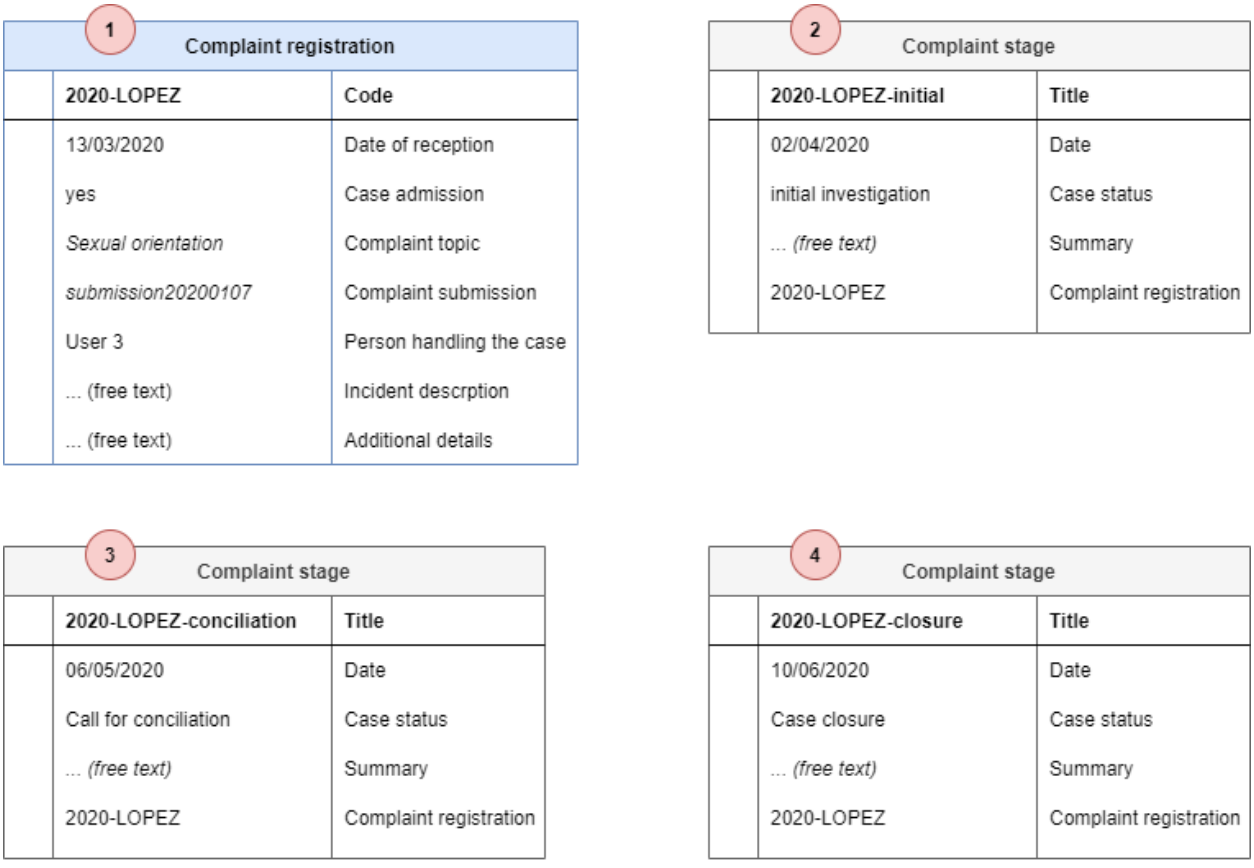


Diagram 2. Data structure with each complaint stage represented as a property

As you can see in the example, the *complaint registration* is linked to the *complaint submission* (the form directly filled by the complainant). The other elements belong to the *complaint stage* type and include the field *case status* to highlight the situation of the complaint in the process. The stage, then, is recorded as a value of the *case status* property and not as an element itself.

This solution was adopted because for this database it was crucial to provide performance indicators such as the complaints on time and overdued, the number of days per stage or the timeline for each complaint. Recording stages this way facilitates the automatic generation of the dashboards that will be presented in the analysis section.

Another potentially effective solution would have been to create a different type of element for each stage, as we present in this example:

1 Complaint registration	
2020-LOPEZ	**Code**
13/03/2020	Date of reception
yes	Case admission
Sexual orientation	Complaint topic
submission20200107	Complaint submission
User 3	Person handling the case
... (free text)	Incident descrption
... (free text)	Additional details

2 Initial investigation	
investigation-2020-LOPEZ	**Title**
02/04/2020	Date
Investigator 3	Investigator
... (free text)	Investigation outcome
Surname, Name	Witnesses interviewed
... (free text)	Evidence description
2020-LOPEZ	Complaint registration

3 Call for conciliation	
conciliation-2020-LOPEZ	**Title**
06/05/2020	Date
... (agreement description)	Conciliation agreement
User 5	Person in charge of verification
... (free text)	Recommendations
2020-LOPEZ	Complaint registration

4 Case closure	
closure-2020-LOPEZ	**Title**
10/06/2020	Date
Conciliation agreement verified	Result of the case
... (free text)	Summary
2020-LOPEZ	Complaint registration

Diagram 3. Data structure with each complaint stage represented as its own entity type

As you can see, this would have allowed us to describe each stage with different properties but, on the other hand, this way it would have been more complicated to show the performance outputs that were very relevant for the partner.

In either case, the model must be completed with two or three more elements: the *complainant*, the *respondent* and, optionally, the *victim*. The *complainant* and the *respondent* are mandatory elements, but the *victim* is not, as this information could be integrated in the *complaint registration* template mentioned before. The parties involved are described with contact details, socioeconomic status, identification data, etc. We present a basic solution for the *complainant* and the *respondent* templates in the following example:

Complainant	
Complainant003	**Code**
Lastname	Last name
Firstname	First name
Streetname, 00. Zipcode. City	Address
+99 999 999 999	Telephone
@	Email
submission20200107	Complaint submission
2020-LOPEZ	Complaint registration
Female	Complainant gender
Respondent034	Respondent
Elderly	Vulnerable group

Respondent	
Respondent034	**Code**
Lastname	Last name
Firstname	First name
Streetname, 00. Zipcode. City	Address
+11 111 111 111	Telephone
@	Email
individual	Type of respondent
submission20200107	Complaint submission
2020-LOPEZ	Complaint registration
Complainant003	Complainant

Diagram 4. Properties on Complainant and Respondent entity types

Collecting information from victims or complainants

As the workflow shows in Diagram 1 above, the process starts with the *complaint submission*, which is directly submitted from an online form by the complainant. The complaint can also be presented by phone or in person. In both cases, the personnel from the organization fills the form with the information provided by the complainant. There is a third option that also needs to be considered: the organization can start a complaint process on its own motion. In the three cases the same complaint submission is filled.

The form has mandatory and optional fields. For instance, a complaint cannot be presented without providing the name and the email of the victim and the complainant. The submission form includes the information that later is included in the *complaint registration*, *complainant* and *respondent* templates.

To facilitate this process, information from matching fields in the *complaint submission* and the *complaint registration* templates can be directly copied and pasted in the system into other templates.

The following example shows what the *complaint submission* (Stage 0) looks like and how its information feeds the next templates:

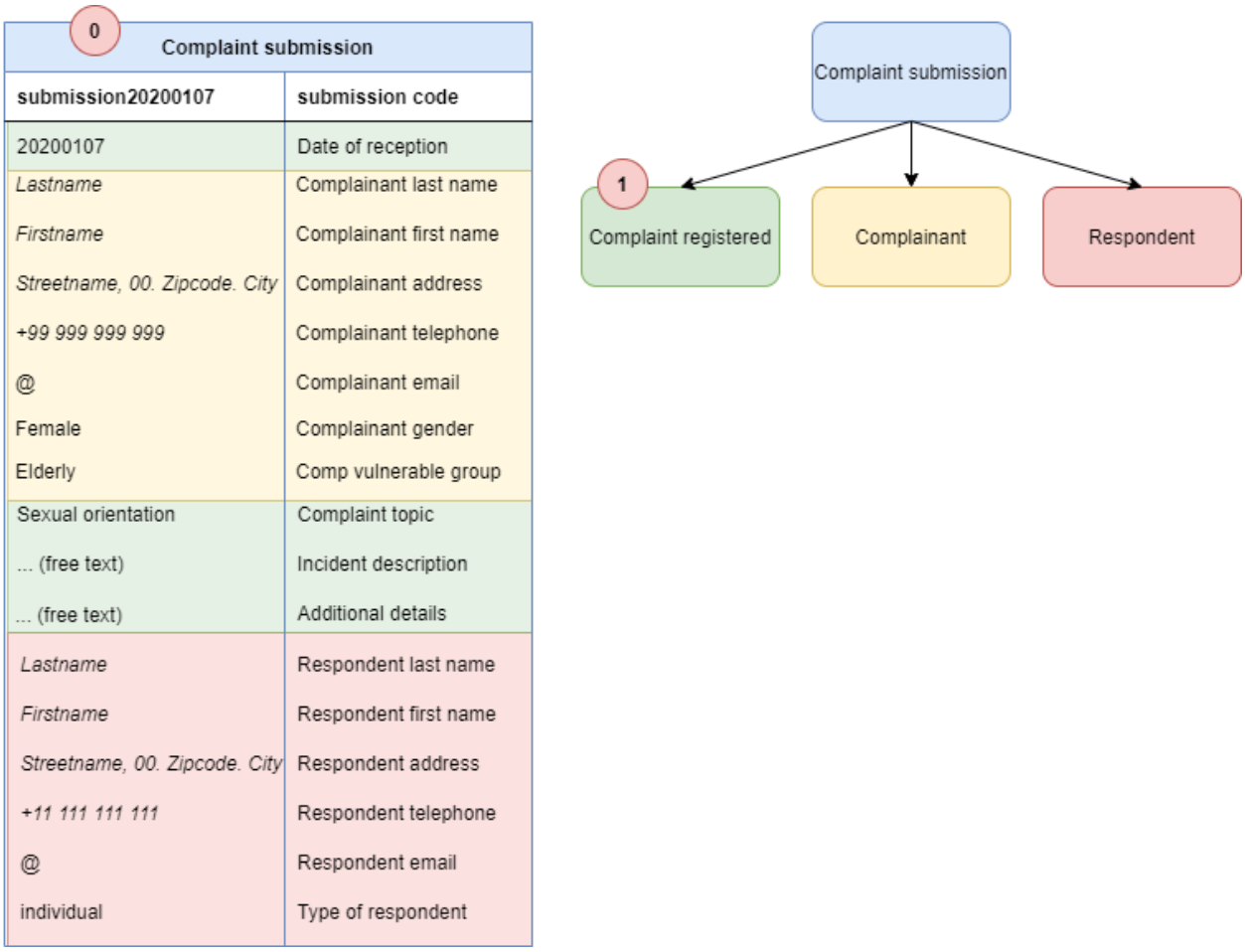


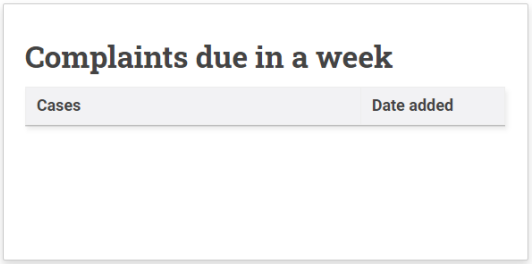
Diagram 5. Properties on Complaint submission and relationships with other entities.

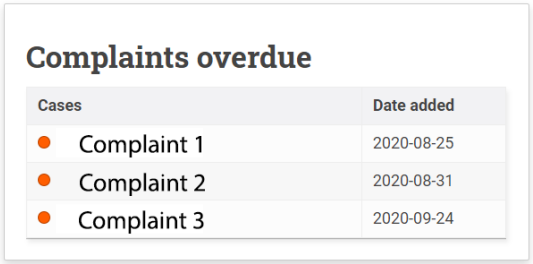
Automate updates for parties

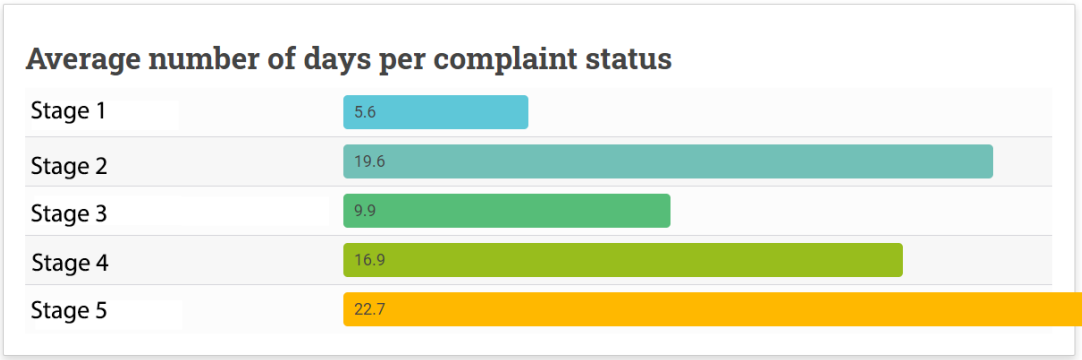
One of the goals of the database was to improve the efficiency of the personnel of the organization by automating repetitive tasks. The best example to show this is the email notification of complainants and respondents. There are specific stages at the process, such as the decision to admit the complaint or to investigate it, that trigger an automatic email for involved parties. This was a customized feature that involved specialized work by developers.

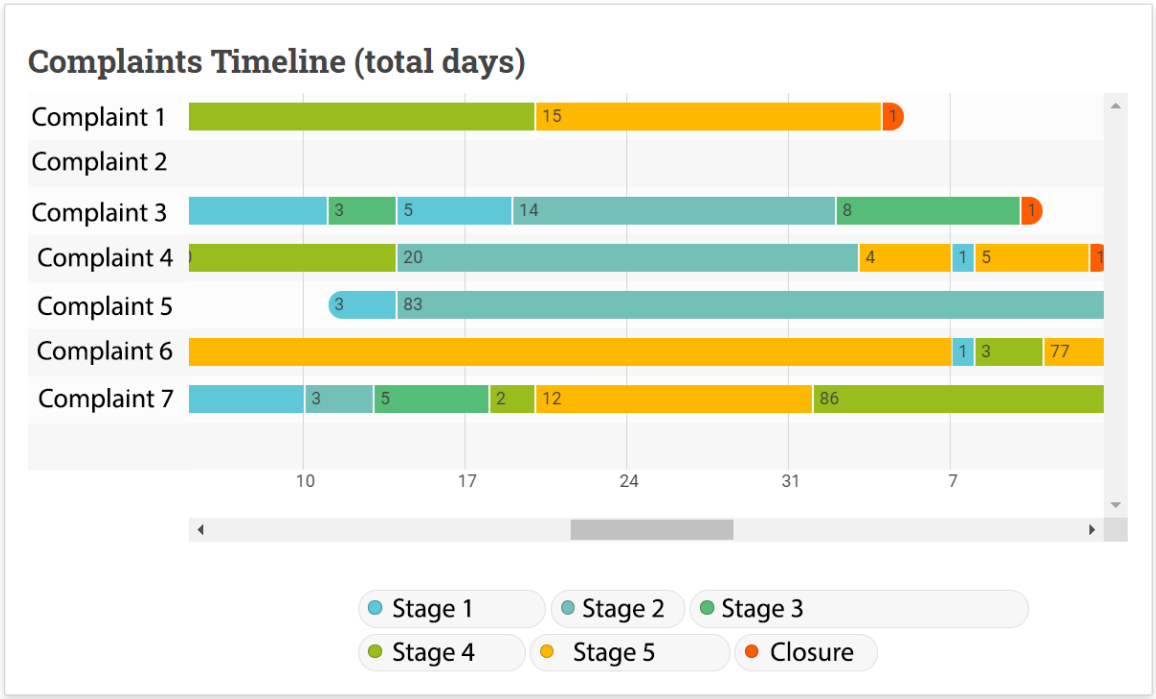
Database outputs

The database allows you to retrieve complaints, complainants and respondents by searching for values within the fields of these entities. Additionally, users can filter results by type of entity. A custom developed dashboard serves as the main tool to visualize the information of the database. These control panels facilitate a quick overview of the most important metrics of the complaint handling process to monitor if the goals are being fulfilled:

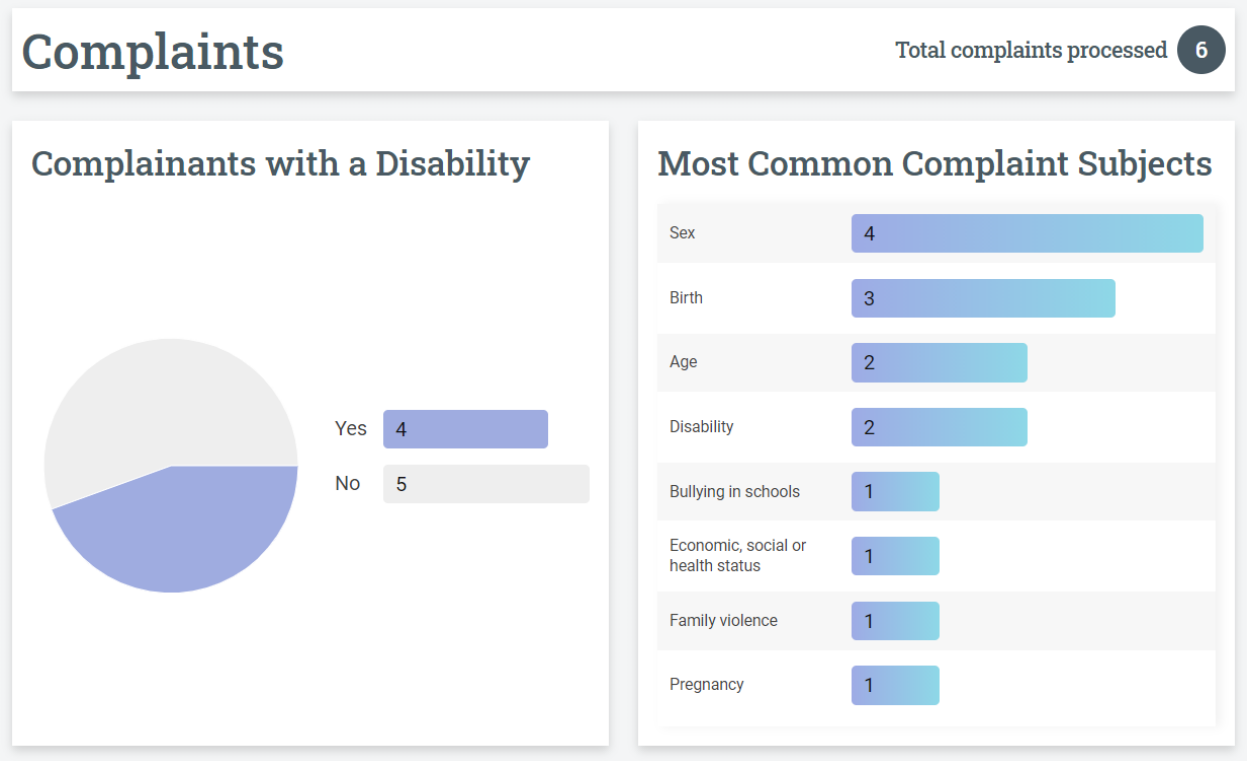







Complaints Timeline (total days)
Complaint 1
15
1
Complaint 2
Complaint 3
3
5
14
8
1
Complaint 4
20
4
1
5
Complaint 5
3
83
Complaint 6
1
3
77
Complaint 7
3
5
2
12
86
10
17
24
31
7
Stage 1
Stage 2
Stage 3
Stage 4
Stage 5
Closure

Graphic information about certain characteristics of the complaint and the complainant is also provided:



Learnings and recommendations

Beyond how sophisticated the tool is and the number of customizations a specific project might include, this case illustrates common human rights documentation needs and ways to address them strategically. These are some of the learnings and recommendations we can take away from the project:

1. **Be able to get a deep understanding of one case, and also get an overview of aggregated data**

   It is common to look at cases or complaints individually, but it is also useful to get a general view about the cases as a whole. In this project, this was addressed by highly customized features providing visualizations and charts.

If the tool you are using does not have that option, make sure you can export your data in interoperable formats, such as CSV, so that you are able to import it into another tool. Rather than having one perfect tool it is worth selecting and combining the right ones.

2. **Tracking the status of a process is a basic task for human rights organizations, and it can be used for multiple purposes.**

   This project was about tracking complaints through the stages of a predefined process, but it is conceptually similar to legal tracking or victim assistance tracking. These purposes have something in common: they are guided by a well-defined workflow in which different paths are clearly identified.

3. **Accountability and performance monitoring demand the anticipated definition of measurable goals.**

   If you want to be notified before overdue dates you need to clearly define complaint handling standards, including the maximum number of days at each stage. This will be useful not only for your information system, but also for your reliability as an organization.

4. **Managing large amounts of data is simpler with automated tasks.**

   If your data is well structured and your tool includes these features, it is possible to automate some tasks. In this project, frequent queries are shown in dashboards, so users don't need to perform them every time they need to extract that information.

   Notifying parties (via email) at different stages of the process was also identified as a time-consuming activity. The system was customized by developers to automate this functionality.

   Even if you are using simple tools such as Excel or LibreOffice Calc, you can use conditional or mathematical formulas to get the most out of your data without doing everything manually.

5. **Integrating the data collection in the workflow saves a lot of time.**

   Having a submission form feeding your database is the most efficient. Nowadays there are plenty of solutions to do this, including Uwazi, but even if you cannot connect them directly, you should be able to download the collected data in an interoperable format, such as CSV, and load it into your database. To facilitate this operation, the collection form and the database structure should be identical.

   In this project, the transmission of information between elements representing different stages of the project was facilitated with a customized copy-paste feature which detected matching fields between templates.