# Introduction

GemsTracker is an application containing (access to) sensitive, privacy protected information. As a result the data protection rules are strict. All access to the data requires authentication and the application has to be able to log what user tried to do for auditing, review and sometimes debugging purposes.

For this reason the routing mechanism of the application is used to provide a central point of access that checks authentication and access rights as well as for logging.

In GemsTracker 1 this resulted in a Gems Menu object that was grown organically over time. As a result it does use the best practices in programming used elsewhere in the application and it is often more complex than needed.

This document tries to present an overview of the current definitions - with some remarks concerning their current suitability –, the describes some use cases and ends with a list of requirements.

# Definitions & Descriptions

## Access Log

GemsTracker has multiple logs. The access log shows who (tried to) access what page and who changed / entered what data.

Currently recorded are:

1. the route accessed,
2. the action performed,
3. the user performing the action,
4. the organization used by that user,
5. the role of that user at that moment,
6. the respondent / patient (if applicably) that is accessed,
7. the screen messages generated by the action,
8. the new data set by that action.

Most of this data is required by the NEN 7513 norm..

## Access Log Setup

Not all routes need to be logged for every entry action. For example: when looking at the current Project Setup, no privacy sensitive is visible and nothing is changed so we need not register this. Some actions, e.g. cleaning the cache also may not merit registration.

Other actions that are not privacy relevant do need to be registered, e.g. changes to the database structure, changes to e.g. organization and especially changes to the access rights themselves.

Lastly logging of most actions regarding respondents / patients, including looking at their data, are required by NEN 7513.

GemsTracker currently uses a Log setup table to determine which actions are logged and which are skipped. This works well for the current purposes, but the system might be improved by an additional table of hard coded actions that are always logged.

## Access Right

An access right is the right to access one or more routes in the application **or** the right to access certain information or access to screen controls. Examples of the former are access to the right to lookup respondents / patients and the separate right to edit them. Examples of the latter are the right switch between multiple organizations or the right to see deleted respondents.

Currently we have chosen to combine access to lookup items with access to see the details of those items. Whether this is a correct combination is a matter of debate, i.e. when rebuilding the system we might choose to implement these right separately.

## Embedded login

Embedded login occurs when another system, e.g. the EPD, shows a GemsTracker screen within a frame in the application of in a pop-up browser window.

The Embedded Login system user has a secret key and several different authentication mechanisms for securely connecting to GemsTracker. After login using the embedded user the system switches to the real user (using the EPD system) so GemsTracker knows this is the user whose eyes are watching the screen.

## Group

Each user is assigned a group and a group is currently assigned a single role. Groups are used to add additional security functionality to users. Groups determine:

1. Which groups the user may assign to other users.
2. IP-ranges that may be used by members if that group with or without 2FA.
3. The screen settings, i.e. how the respondent search / show / edit screens look.
4. Masking for privacy sensitive information, e.g. hiding email addresses and birthdays.
5. Surveys are assigned to a certain group for answering (in survey maintenance).
6. Answer visibility van also be limited to certain groups (in survey maintenance).

While a user has (currently) only a single group, there are two cases where the actual group that is used is different from the users group. One is when the user switches to a different group manually, usually for checking the rights of another group.

The second case is during Embedded Login, where we usually use another group that show more limited information and allows less actions than during a normal login.

## Module

Since version 1.9.0 GemsTracker allows extension of functionality using modules.

Modules access the Menu object and add extra routes and rights to it, but use the Menu object as well in order to generate url’s and and add buttons in the interface.

## Organization

## Respondent / Patient

## Roles

A role is a list of access rights assigned a single label, like ‘staff’ or ‘admin’. Roles are also hierarchical, i.e. the admin role contains all rights of the staff roles *at the time the admin roles was edited!* I.e. if a staff user could edit a patient when the admin roles was edited, the admin keeps this right.

This hierarchy is a historical left-over from the original setup of the application. Removing the hierarchy for the roles and assigning multiple roles to groups instead might be better a solution.

There are three hard coded roles in the application.

### No Login

The No Login role contains the rights assigned to anyone not logged in. So it contains the right to access the Login page and e.g. the Contact page, but not to Logoff page and the page to look up respondents.

### Respondent

The respondent role is a role that can be assigned to respondents to allow respondent login and determines the pages they can see.

### Master

The Master role is created by the role system itself and always contains all defined rights, with the exception of the login page itself. It is assigned by default to the project user and is used during initial setup as well as when some error in the rights system makes the application inaccessible for other users.

## Route

## Sites

## Status

## Token

## User

### No Login

### Patient

### Project user

### Respondent

### System users

# Use cases

# Requirements