BIRDWATCH

The Diversity of Life from Space

D6.1 BirdWatch Backend Database

Deliverable for the Horizon Europe Project BirdWatch

V1.0

Legal Disclaimer

Funded by
the European Union

Document Information

GA Number	101082634	**Type of Action**	Horizon-IA
Full Title	BirdWatch - a Copernicus-based service for the improvement of habitat suitability of farmland birds via satellite-enabled monitoring, evaluation and optimisation of CAP greening measures		
Project Acronym	BirdWatch		
Start Date	February 1st, 2023	**Duration**	36 Months
Project URL	https://birdwatch-europe.org/		
Deliverable	D6.1: BirdWatch Backend Database		
Work Package	WP6		
Project Month of Delivery	**Contractual** M13	**Actual**	M13
Nature	OTHER	**Dissemination Level**	PU
Lead Beneficiary	LUP		
Responsible Author	Nastasja Scholz, LUP		
Contributions from	Pierre Babeck, LUP Stefan Braumann, LUP Jocelyn Knight, LUP Annett Frick, LUP		

Funded by the European Union

History of Changes

Version	Issue Date	Stage	Description	Comments	Contributor
1.0	28.02.2024	Draft	First draft of D6.1		Nastasja Scholz - LUP

Funded by
the European Union

Table of Contents

Funded by
the European Union

Introduction

BirdWatch's aim is to provide an EU-wide service supporting the monitoring and improvement of farmland habitat suitability for bird species which breed or forage on agricultural land.
The BirdWatch service will consist of a geospatial data-based monitoring service which evaluates the habitat suitability of farmland parcels for specific bird species as well as of an optimisation workflow, serving as a decision-support for the identification of appropriate policy-supported agri-environmental measures. The geospatial data will partly consist of environmental parameter data derived from the freely available satellite imagery from Sentinel-1 and Sentinel-2 of the Copernicus Program of the European Space Agency (ESA) and auxiliary spatial datasets, most importantly the national or regional Land Parcel Identification System (LPIS) data.

The service will be made available to users via a web-based GIS application. The heart of this GIS application will be the backend database where all necessary data storage and management will take place. This includes the data components necessary to calculate the state of habitat suitability and its temporal evolution as well as the various constraints and the associated optimised habitat suitability, provided by VITO's MooV service. The web-based user interface will also access this database for providing the results to the users.

The database scheme, as it is described in this deliverable, was created using the input data and example results for our test region in Flanders, the selection of environmental parameters with which to build the habitat models as well as the foreseen ways stakeholders will be able to interact with the BirdWatch platform.
The deliverable thus describes the first version of the backend database, but changes following first implementation tests are likely, especially since the selection of environmental parameters is still ongoing.

Funded by
the European Union

Requirements of the Backend Database

The backend database is designed with the following *data requirements* (also see *D2.4 - User and System Requirements*) in mind.
The database should store

- raster-based optical, radar and thermal satellite imagery from sources such as Copernicus Sentinel- or the Landsat missions.
- a background grid database which serves as the reference grid for the parameter derivation and habitat suitability calculation.
- information on land use types, crop types, soil types, administrative boundaries, protected areas, etc.
- the calculated habitat suitability in vector-, raster- and tabulated format, as derived by the platform.
- the bird observation data with information on species, location, date, number of individuals, etc.
- tabular data with bird-specific and stakeholder-specific constraints.
- user data (e.g., email address, username, encrypted passwords, etc.)
- in the near future: weather data such as temperature, precipitation, wind speed, etc. from sources such as meteorological stations or weather APIs.

The following requirements were also taken into account:

- data interoperability (e.g., vector data needs to be rasterised, raster data vectorised)
- required storage per test region
- data security and data backup requirements

Funded by
the European Union

Description of the Backend Database

The backend database is the central storage of the following BirdWatch-related data formats:

- Raster Data (e.g., .geotif, .jp2, .nc)
- Vector Data (.shp, .geojson, .gpkg)
- Tabular Data (e.g., .csv, .xls)
- Text Data (e.g., .txt, .doc)

For the web-based map application, GeoServer is used, which is java-based and allows the handling and display of geospatial data as well as, e.g., data sharing and editing. To enable exchange with external applications, GeoServer uses WMS (Web Map Services) or WFS (Web Feature Services).

The GeoServer will not be accessible to anyone but the administrative personnel.

GeoServer can be connected to a PostGIS Database, which provides functionalities which are important for BirdWatch:

- Spatial data storage
- Spatial indexing
- Functions for spatial data manipulation (buffering, intersecting, etc.)
- Geospatial processing (e.g., conversion)
- Integration, e.g., into QGIS

PostGIS is based on PostgreSQL, a performative open source object-relational database system.

How the database will be integrated into the system architecture of the BirdWatch platform is visualised in the following figure.

Funded by
the European Union

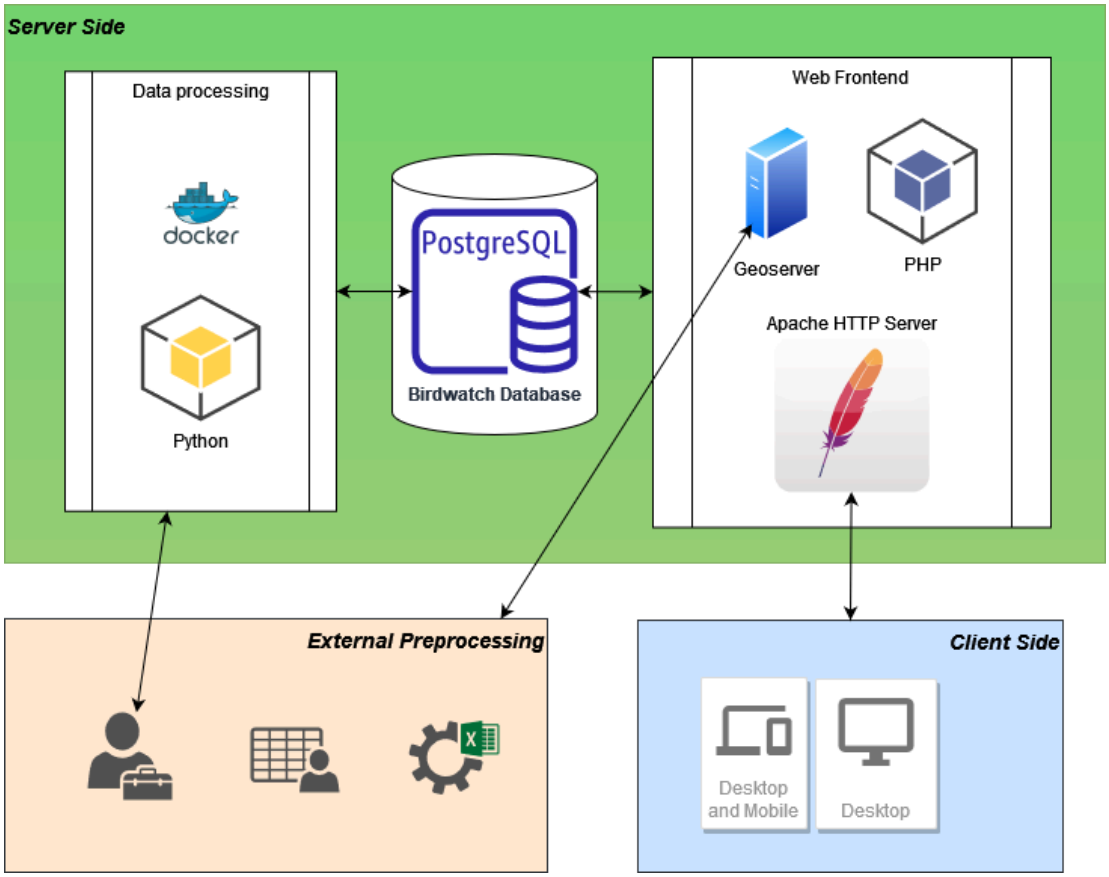

Fig. 1: BirdWatch platform system architecture

The following technologies for the implementation of server and database components are planned:

- Server Linux: AlmaLinux9
- Apache HTTP Server 2.4
- PHP 8.3
- PostgresSQL 13 / PostGIS 3
- GeoServer 2.25
- Python 2.X and Python 3.X
- docker 4.X

The Web-GUI is currently planned with the following components:

- HTML5
- CSS3
- jQuery 3.7

Funded by
the European Union

- Bootstrap 4.6
- OpenLayers 6.X

This deliverable focuses on the BirdWatch Database and the internal storage. Data exchange and its operationalisation with external service providers, i.e., with Sentinelhub[1], openEO[2] and MooV[3] (visualised as "External Preprocessing" in the figure above), still have to be defined and will be based on lessons learned during testing.

In the following the database schema is explained in more detail.

[1] https://www.sentinel-hub.com/
[2] https://openeo.org/
[3] https://moov.vito.be/en

Funded by
the European Union

General considerations

All data will be required to be stored with a unique identifier to ensure findability and support interoperability. In addition, all data will have the attributes `changed_on` and `changed_by`, in order to trace any changes to the data.

For each data type, we use PostGIS tables which determine which attributes are required for the data type.
An important decision to make is the use of naming conventions and schemata for the storage of the various types and roles of data.

The prefix of the table names consists of the type of information stored in the table, i.e.:

- `dat_user` → user data
- `def_` → definitions
- `geo_` → geospatial input data and intermediate results
- `lookup_` → lookup tables
- `result_` → output of the habitat suitability calculation and habitat suitability optimisation

The storage of *data* will follow the following nomenclature:
`object name = <prefix>_<name>_<suffix>`

Currently, we only foresee the prefixes `dat_user`, `geo_`and `result_`. This might change with the experiences and feedback from the activities in the test regions.
The suffixes will be used to identify the type, format , content and versioning of the data. The format should tell us if the data is raster-based, i.e., aggregated to the 200 m x 200m EUROSTAT grid[4], or parcel-based, aggregated to the LPIS parcel boundaries.

As an example, soil moisture raster data for the year 2018 will be named:
`geo_cell_based_soilmoisture_2018.tif`
Parcel-based soil moisture data will be named:
`geo_parcel_based_soilmoisture_2018.tif`

Please note that the test region's name is not explicitly mentioned in the nomenclature as it is planned to create one database per BirdWatch region, with the database schemes otherwise remaining the same (i.e., one database for Flanders, Lithuania, South Tyrol, and one for each German federal state).

[4] https://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/grids

Funded by
the European Union

User data

The following figure shows how user data will be stored in the backend database.

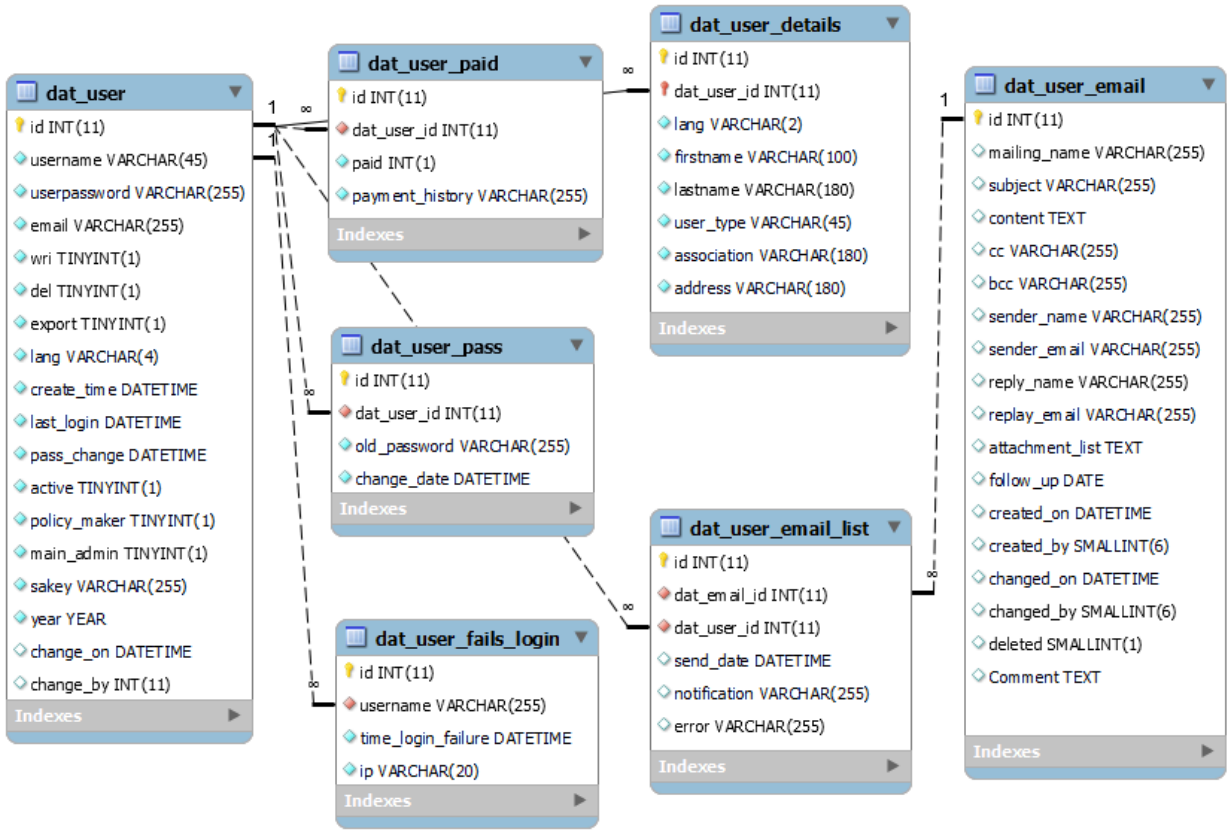

Fig. 2: Scheme of how user data will be stored in the database

`dat_user` is the base table for the user of the system and contains information on the user account and user's rights.

- `username` should not be Zero and should be unique
- `userpassword` is the password a user has set; it has to be encrypted;
- `email` is the email address of a user;
- `wri`, `del`, and `export` are user rights
- `lang` is the language setting of the user (available languages are defined in the table `lookup_languages`)
- `create_time`, `last_login` and `pass_change` respectively store the date and time of creation, last login and change date of the password

Funded by
the European Union

- `active` determines with 0 or 1 whether a user account is active
- `policy_maker` determines whether a user can edit agri_environmental_measures (see table `def_policy_measure`)
- `main_admin` determines whether a user is an administrator and can manage user accounts
- `sakey` stores a temporary activation key for a user

`dat_user_details` contains details about the referenced table `dat_user`.

- it is linked to the parent table via `dat_user_id`
- is stores the language preference via `lang`
- `firstname` and `lastname` store the user's full name
- `user_type` stores the type of the user (e.g., admin, policymaker, farmer, etc.)
- `association` stores the organisation's name to which the user belongs to
- `address` stores the user's address (company or organisation address, if applicable, or address of the farm, if the user is a farmer)
- `user_type`, `association` and `address` can all hold multiple values, as the user can change address or organisation or even his user type (e.g., a former farmer takes up consulting at a farmers organisation)

`dat_user_email_list` contains a list of emails that a user has received from the system.

- it is linked to the specific user table via `dat_user_id`
- `sent_date` contains when it was sent (date and time)
- `notification` contains - if applicable - the type of notification
- `error` contains - if applicable - the type of error message of the system

`dat_user_email` contains the email templates for sending messages regarding, e.g., forgotten password, notifications of account activation, etc.

- it is linked to the specific user table via `dat_user_id`
- it is linked to the `dat_user_email_list` table via `dat_email_id`
- `mailing_name` contains the name of the user with which the user will be addressed in the email
- `subject` contains the subject of the email
- `content` contains the content of the email
- `cc` and bcc contain respective email addresses in case a copy or a blind copy of the email was sent
- `sender_name` contains name of the sender of the email
- `reply_name` contains name of the sender of the email reply
- `attachment_list` contains the list of attachments which was sent, if applicable

Funded by
the European Union

- `follow_up` contains the date of the reply
- `created_on` contains the datetime of the email template
- `created_by` refers to who has created the email template
- `deleted` refers to the deletion of a template
- `Comment` contains any comments

`dat_user_pass` contains the encrypted passwords of a user. It is planned to store up to three old user passwords. A user may not reuse passwords in this table.

- it is linked to the specific user table via `dat_user_id`
- `old_password` contains the old password
- `change_date` contains the datetime of the change of the password

`dat_user_fails_login` contains the details of a failed login attempt. It is also used to coordinate the blocking of an account. If a user logs in with an incorrect password, this will be saved. After trying to log in several times without success, the user account can be blocked and the IP address blocked for security reasons.

- it is linked to the specific user table via `username`
- `time` contains the datetime of the failed login attempt
- `ip` contains the IP address from which the failed login has been created

`dat_user_paid` is planned to be used for the information regarding the users who pay for the service.

- it is linked to the specific user table via `dat_user_id`
- `paid` signifies if the user has paid for the licence
- `payment_history` stores information on previous payments

Funded by
the European Union

Lookups and definitions

Lookup tables are used for associating general information, e.g., a list of countries and institutions. Definition tables store BirdWatch-specific information, e.g., the list of target species and the regions for which they are a relevant species.

The following figure shows how lookup tables will be stored in the backend database.

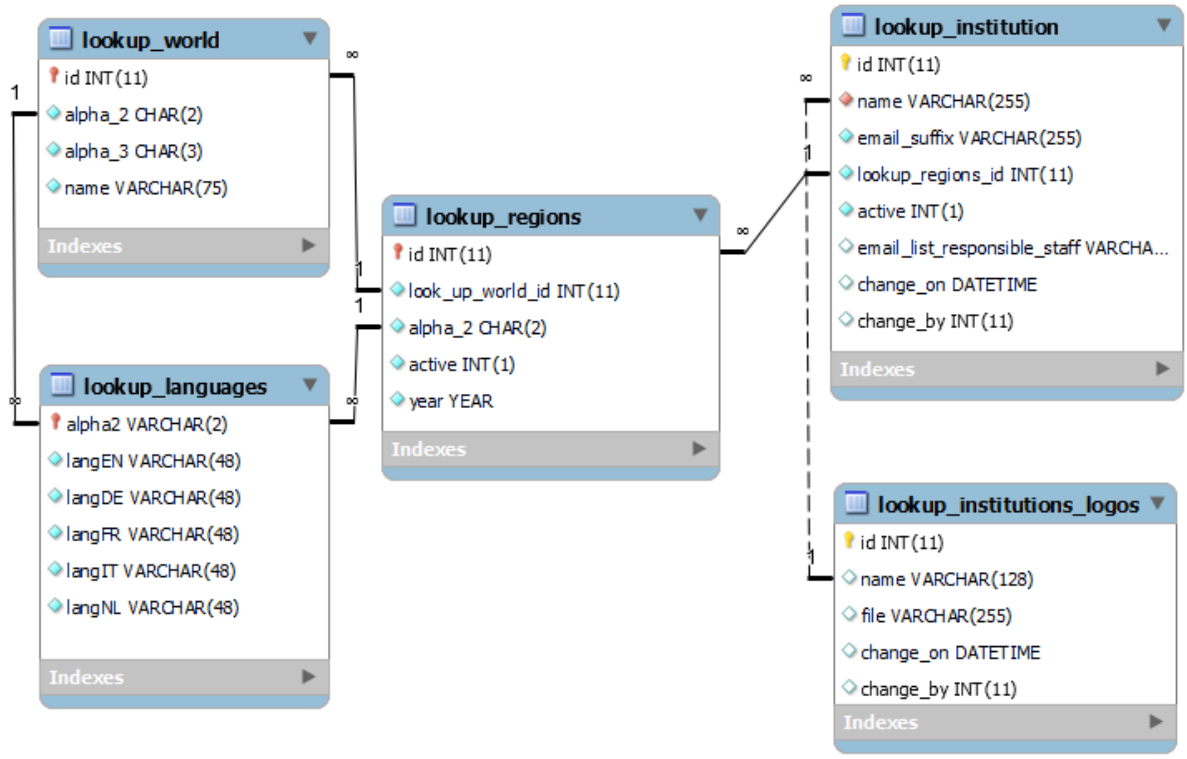

Fig. 3: Scheme on how lookup tables will be stored in the database

`lookup_world` contains the list of all countries in the world.

- `alpha2` represents the second letter of the country code for the country
- `alpha3` represents the third letter of the country code for the country
- `name` is the full country name

`lookup_regions` contains the list regions in which BirdWatch is active.

- `alpha2` represents the second letter of the country code for the country
- `year` refers to the year in which the region was added to BirdWatch
- `active` indicates if the BirdWatch service is active in the region

Funded by
the European Union

lookup_languages contains languages which are available in BirdWatch

- it is linked to the table `lookup_world` via `alpha2`
- `langDE` name of the country in German
- `langEN` name of the country in English
- `langFR` name of the country in French
- `langIT` name of the country in Italian
- `langNL` name of the country in Dutch

lookup_institution contains the list of all institutions which have rights to edit the measures (see definitions of measures above).

- `name` name of the institution
- `email_list_responsible_staff` is an optional list of the emails of all staff members in the institution which can edit the measures on the BirdWatch platform
- `active` indicates if the institution can currently edit measures on the BirdWatch platform

lookup_institution_logos contains the list of all institutions which have rights to edit the measures (see definitions of measures above).

- `name` name of the institution to which the logo is linked
- `file` is the path to where the logo is stored

The following figure shows how definitions will be stored in the backend database.

Funded by
the European Union

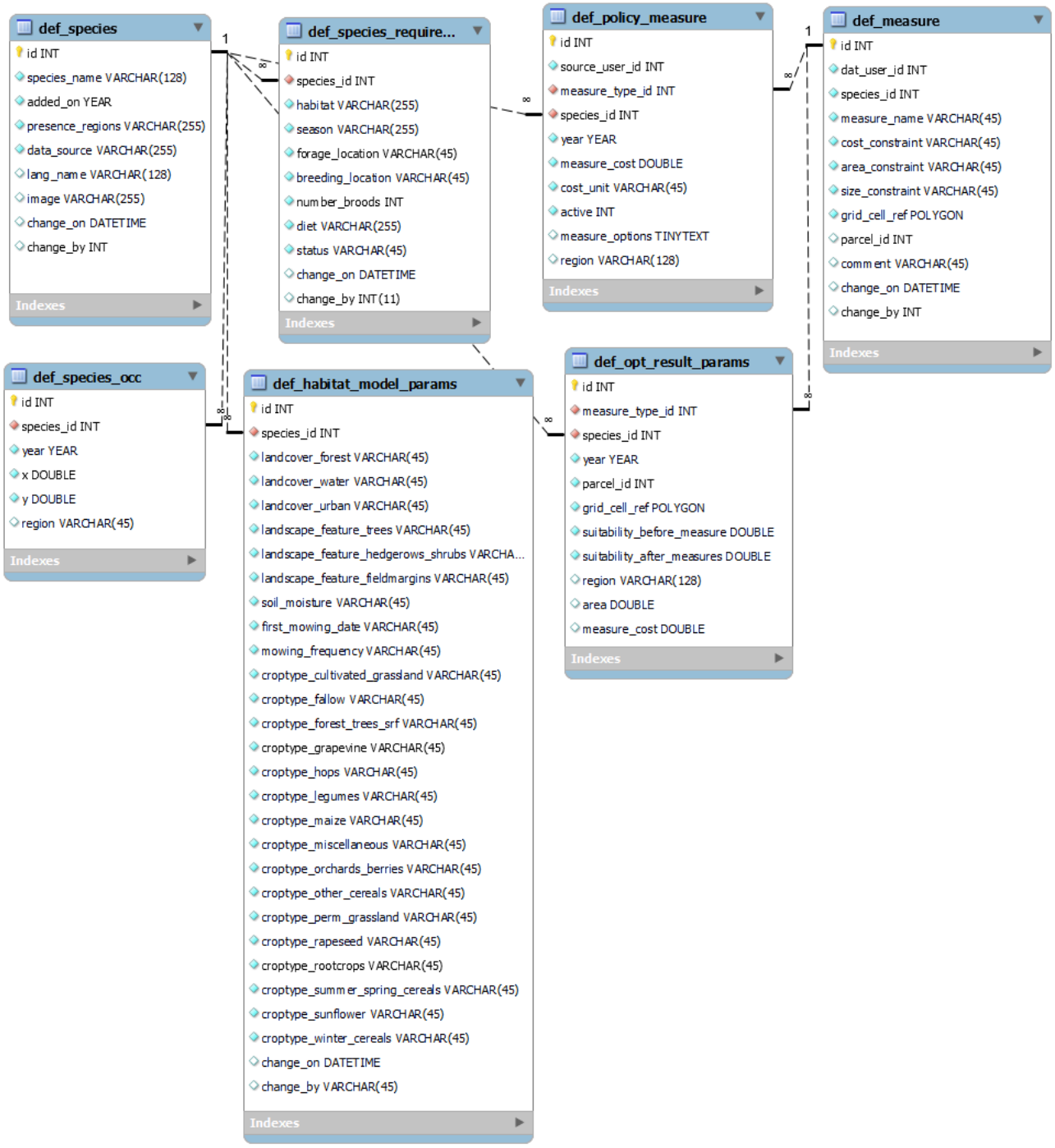

Fig. 4: Scheme of how definitions will be stored in the database

Funded by
the European Union

def_species is the base table for our target species relevant in the region.

- species_name contains the species name of the specific species
- lang_name contains the name of the specific species in the regional language (available languages are defined in the table lookup_languages)
- image links to an image of the species
- add_on contains when the specific species was added to the BirdWatch service
- presence_regions contains the regions in which the species is relevant
- data_source indicates the origin of the data (e.g., DDA)
- lang_name contains the name of the specific species in the regional language (available languages are defined in the table lookup_languages)
- image links to an image of the species

def_species_occ refers to the necessary information regarding the occurrence data of a species.

- it is linked to the specific species via species_id
- year contains the year the occurrence data was collected
- x contains the x-coordinate (UTM) where the occurrence data was collected
- y contains the y-coordinate (UTM) where the occurrence data was collected
- region contains the region in which the occurrence data was collected (optional, since the database is region-bound)

def_species_requirements refers to the information on the species-specific requirements regarding the occurrence data of a species.

- it is linked to the specific species via species_id
- habitat contains information on what habitats the bird prefers (e.g., mosaicked, open)
- season refers to the relevant season for the species in the region (i.e., mostly the breeding season)
- forage_location refers to where the bird likes to look for food
- breeding_location refers to where the bird likes to build its nest (e.g., on the ground, in trees)
- number_broods refers to the number of times a bird species breeds per year
- diet refers to preferred food items
- status refers to the status of endangerment in the region

def_habitat_model_params refers to the necessary parameters required for the habitat suitability calculation.

- it is linked to the specific species via def_species_id

Funded by
the European Union

- `landcover_forest` refers to environmental parameter reflecting the percentage of the landcover consisting of forest
- `landcover_water` refers to environmental parameter reflecting the percentage of the landcover consisting of water
- `landcover_urban` refers to environmental parameter reflecting the percentage of the landcover consisting of urban areas
- `landscape_feature_trees` refers to environmental parameter reflecting the occurrence of landscape features consisting of trees
- `landscape_feature_hedgerows_shrubs` refers to environmental parameter reflecting the occurrence of landscape features consisting of hedgerows or shrubs
- `landscape_feature_fieldmargins` refers to environmental parameter reflecting the occurrence of landscape features consisting of field margins
- `soil_moisture` refers to environmental parameter reflecting the soil moisture derived via remote sensing
- `first_mowing_date` refers to environmental parameter reflecting the first day of mowing per year, derived via remote sensing
- `mowing_frequency` refers to environmental parameter reflecting the mowing frequency of a specific year, derived via remote sensing
- `croptype_cultivated_grassland` refers to environmental parameter reflecting the crop type *cultivated grassland*
- `croptype_fallow` refers to environmental parameter reflecting the crop type *fallow*
- `croptype_forest_trees_srf` refers to environmental parameter reflecting the crop type *Forest / Trees / SRF*
- `croptype_grapevine` refers to environmental parameter reflecting the crop type *grapevine*
- `croptype_hops` refers to environmental parameter reflecting the crop type *hops*
- `croptype_legumes` refers to environmental parameter reflecting the crop type *legumes*
- `croptype_maize` refers to environmental parameter reflecting the crop type *maize*
- `croptype_miscellaneous` refers to environmental parameter reflecting the crop type *miscellaneous*
- `croptype_orchards_berries` refers to environmental parameter reflecting the crop type *orchards and berries*
- `croptype_other_cereals` refers to environmental parameter reflecting the crop type *other cereals*
- `croptype_perm_grassland` refers to environmental parameter reflecting the crop type *permanent grassland*

Funded by
the European Union

- `croptype_rapeseed` refers to environmental parameter reflecting the crop type *rapeseed*
- `croptype_rootcrops` refers to environmental parameter reflecting the crop type *root crops*
- `croptype_summer_spring_cereals` refers to environmental parameter reflecting the crop type *summer/spring cereals*
- `croptype_sunflower` refers to environmental parameter reflecting the crop type *sunflower*
- `croptype_winter_cereals` refers to environmental parameter reflecting the crop type *winter cereals*

`def_policy_measure` refers to a policy-backed farmland management measure (i.e., an agri-environmental measure) applicable in the region; this input comes from an authority of the region (e.g., policymaker, paying agency, etc.);

- it is linked to the specific species via `species_id`
- `source_user_id` refers to the unique identifier of the user who has put in the details (i.e., usually a policymaker)
- `measure_type_id` refers to the unique identifier of the measures type (e.g., mowing related measures)
- `year` refers to the year the information on the measure was put in
- `active` indicated is this measure is currently applicable
- `measure_cost` contains information on the budget related to the measure
- `cost_unit` contains information on the unit to which the cost refers (e.g., EURO/ ha)
- `measure_options` contains optional information on the measure
- `region` contains the region in which the measure is applicable

`def_measure` refers to input measure to be optimised

- it is linked to the table `def_policy_measure` via `measure_type_id`
- `dat_user_id` refers to the associated user who defined the constraints (e.g., a specific farmer)
- `measure_name` refers to the name of the measure
- `cost_constraint` refers to budget constraints
- `area_constraint` refers to spatial constraints
- `size_constraint` refers to constraints to what extent the measure is carried out (e.g., limitation of number of trees to plant)
- `grid_cell_ref` refers to the associated grid cell in the EUROSTAT grid (see `geo_reference_cell_grid`)
- `parcel_id` refers to the associated parcel (see `geo_parcel_data`)

Funded by
the European Union

def_opt_result_params refers to the necessary output parameters of the optimisation

- it is linked to the specific species via species_id
- it is linked to the specific policy measure via measure_type_id
- year refers to the year for which the optimisation is calculated
- grid_cell_ref refers to the associated grid cell in the EUROSTAT grid (see geo_reference_cell_grid)
- parcel_id refers to the associated parcel (see geo_parcel_data)
- suitability_before_measure refers to the initial habitat suitability value
- suitability_after_measure refers to the habitat suitability value when optimised measure would be applied
- area contains the area size affected by the optimisation
- measure_costs contains the associated cost of the optimised measure
- size refers to the extent the measure is carried out
- region contains the region

Funded by
the European Union

Input data, intermediate and final results

Geodata tables refer to the input data and intermediate results, the results tables refer to the final results, i.e., the calculated and optimised habitat suitability.

The following figure shows how geodata will be stored in the backend database.

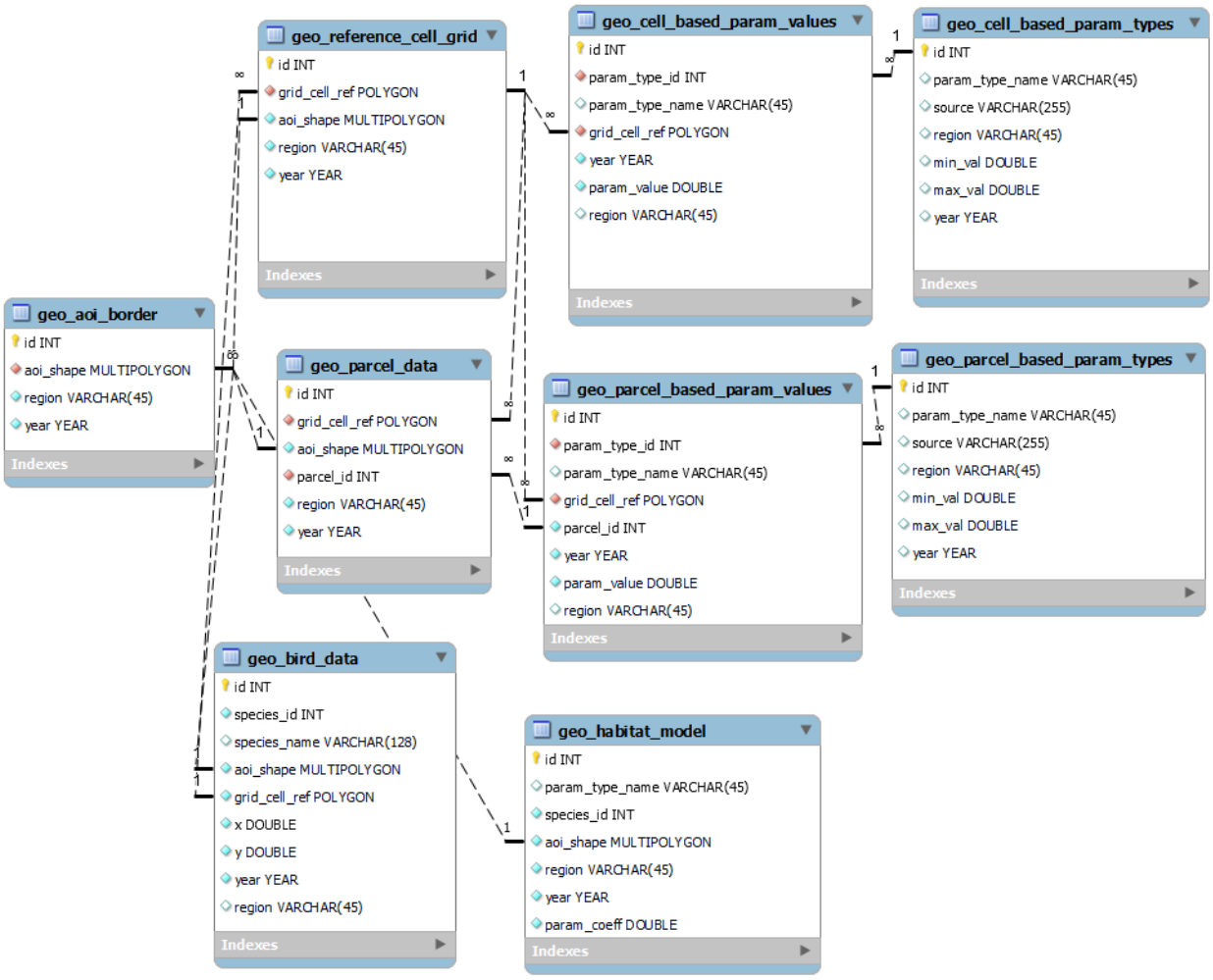

Fig. 5: Scheme of how geodata will be stored in the database

Funded by
the European Union

geo_aoi_border contains the boundary of the region (i.e., AOI).

- `aoi_shape` refers to the boundary coordinates of the AOI
- `region` is the name of the AOI (e.g., "Flanders")
- `year` is the year in which the vector file of the AOI was added

geo_reference_cell_grid contains the reference EUSTAT Grid covering the region.

- `grid_cell_ref` contains the coordinates of the grid
- it is connected to the aoi via `aoi_shape`
- `region` is the name of the region (e.g., "Flanders")
- `year` is the year in which the vector file of the grid was added

geo_parcel_data contains the LPIS-based parcel data for the region.

- `parcel_id` is the parcel ids in the file;
- it is connected to the aoi via `aoi_shape`
- it is connected to the reference grid via `grid_cell_ref`
- `region` is the name of the AOI covered by the LPIS data (e.g., "Flanders")
- `parcel_shape` contains the coordinates of the parcels in the file
- `year` is the year in which the LPIS file was added

geo_bird_data contains the bird species observation data for the region .

- `species_id` signifies which bird species is represented in the data
- `species_name` contains the species name of the specific species
- it is connected to the aoi via `aoi_shape`
- it is connected to the reference grid via `grid_cell_ref`
- `year` contains the year the occurrence data was collected
- `x` contains the x-coordinate (UTM) where the occurrence data was collected
- `y` contains the y-coordinate (UTM) where the occurrence data was collected
- `region` contains the region in which the occurrence data was collected

geo_habitat_model contains the habitat models for each bird species relevant for a region.

- `species_id` signifies which bird species is represented in the data
- it is connected to the aoi via `aoi_shape`
- `year` contains the year the occurrence data was collected
- `region` contains the region in which the occurrence data was collected
- `param_type_name`, here the placeholder for the respective name of each of the sdm_parameters listed in `def_habitat_model_params`;
- `param_coeff`, here the placeholder for the respective value of each of the sdm_parameters listed in `def_habitat_model_params`;

Funded by
the European Union

geo_cell_based_param_values contains the raster data for each of the environmental parameters.

- `param_type_id` refers to the parameter in `def_habitat_model`
- `param_type_name`, here the placeholder for the respective name of each of the sdm_parameters listed in `def_habitat_model_params`;
- `param_value` contains the parameter value
- `year` is the year for which the raster data is input for the habitat suitability calculation
- it is connected to the reference grid via `grid_cell_ref`

geo_cell_based_param_types contains information on the environmental parameter stored in raster data.

- `param_type_name` is the name of the habitat model parameter the file contains
- `source` is the source of the parameter (e.g., Sentinel2-based)
- `region` is the name of the region (e.g., "Flanders")
- `min_val` and `max_val` are the minimum and maximum allowed values for this parameter
- `year` is the year for which parameter is input for the habitat suitability calculation

geo_parcel_based_param_values contains the LPIS-based vector data for each of the environmental parameters.

- `param_type_id` refers to the parameter in `def_habitat_model`
- `param_type_name`, here the placeholder for the respective name of each of the sdm_parameters listed in `def_habitat_model_params`;
- `param_value` contains the parameter value
- `year` is the year for which the raster data is input for the habitat suitability calculation
- it is connected to the reference grid via `grid_cell_ref`
- it is connected to a specific parcel grid via `parcel_id`

geo_parcel_based_param_types contains information on the environmental parameter stored in the LPIS-based vector data.

- `param_type_name` is the name of the habitat model parameter the file contains
- `source` is the source of the parameter (e.g., Sentinel2-based)
- `region` is the name of the region (e.g., "Flanders")
- `min_val` and `max_val` are the minimum and maximum allowed values for this parameter
- `year` is the year for which parameter is input for the habitat suitability calculation

Funded by
the European Union

The following figure shows how habitat suitability results will be stored in the backend database.

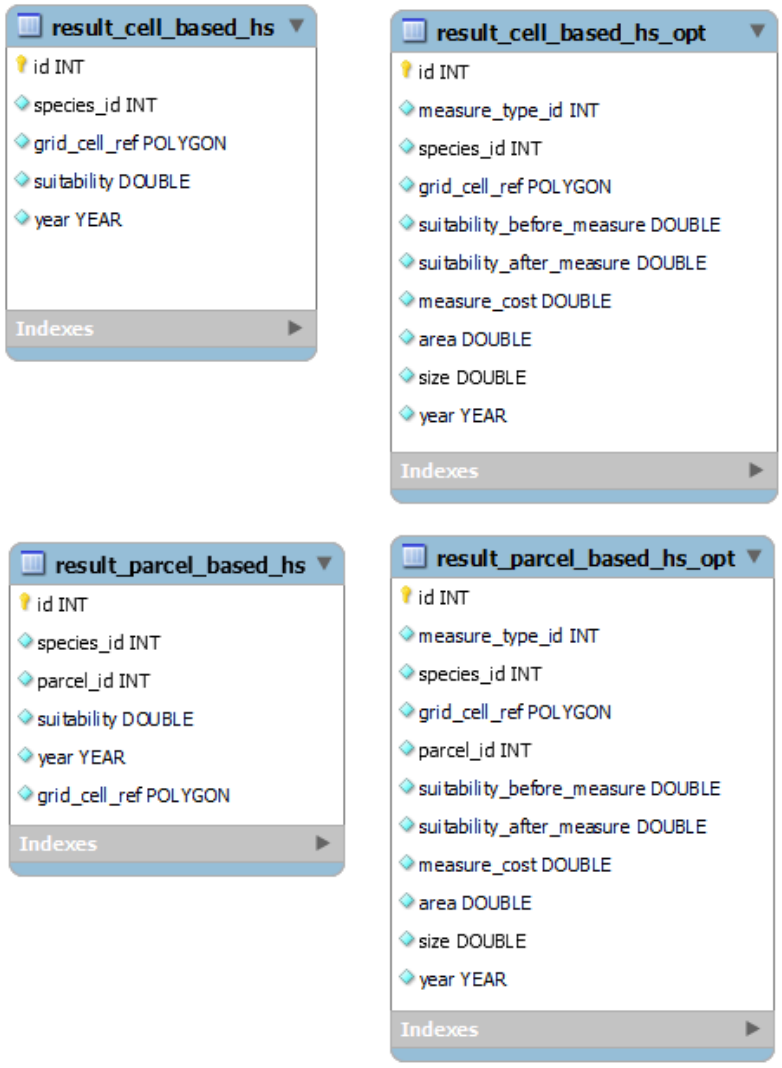

Fig. 6: Scheme of how habitat suitability will be stored in the database

`result_cell_based_hs` contains the raster-based habitat suitability for a specific bird species.

- `species_id` signifies the relevant bird species
- it is connected to the reference grid via `grid_cell_ref`
- `suitability` is the resulting habitat suitability

Funded by
the European Union

- `year` is the year for which the habitat suitability was calculated

`result_parcel_based_hs` contains the raster-based habitat suitability for a specific bird species.

- `species_id` signifies the relevant bird species
- `parcel_id` refers to the associated parcel
- it is connected to the reference grid via `grid_cell_ref`
- `suitability` is the resulting habitat suitability
- `year` is the year for which the habitat suitability was calculated

`result_cell_based_hs_opt` contains the raster-based optimised habitat suitability for a specific bird species.

- `species_id` signifies the relevant bird species
- it is connected to the reference grid via `grid_cell_ref`
- it is linked to the specific policy measure via `measure_type_id`
- `year` refers to the year for which the optimisation is calculated
- `suitability_before_measure` refers to the initial habitat suitability value
- `suitability_after_measure` refers to the habitat suitability value when optimised measure would be applied
- `area` contains the area size affected by the optimisation
- `measure_costs` contains the associated cost of the optimised measure
- `size` refers to the extent the measure is carried out
- `region` contains the region

`result_parcel_based_hs_opt` contains the vector-based optimised habitat suitability for a specific bird species.

- `species_id` signifies the relevant bird species
- it is connected to the reference grid via `grid_cell_ref`
- it is linked to the specific policy measure via `measure_type_id`
- `year` refers to the year for which the optimisation is calculated
- `parcel_id` refers to the associated parcel
- `suitability_before_measure` refers to the initial habitat suitability value
- `suitability_after_measure` refers to the habitat suitability value when optimised measure would be applied
- `area` contains the area size affected by the optimisation
- `measure_costs` contains the associated cost of the optimised measure
- `size` refers to the extent the measure is carried out
- `region` contains the region

Funded by
the European Union

Moving forward

The database described in this deliverable is a representation of how we currently plan to store and manage data.
The backend database will be created in the upcoming months, be filled with the data for our first test region, i.e., Flanders, and integrated into our first implementation of the web-based platform (D6.2, M18). The practically of our database design will be tested together with the first version of the web-based platform.

Some required changes are likely to occur, especially with regards to the environmental parameters (`def_habitat_model_params`), following the evaluation of their importance as descriptors for habitat suitability (this will be reflected in D4.2 - Ensemble and Joint Species Distribution Models for first Study Area and D4.3 - Ensemble and Joint Species Distribution Models for all Study Areas).

Regional differences in data structures and content will also have to be explored, once we collected all necessary input for our four test regions. Depending on the outcome, this will have to be addressed in an updated database design and reflected in the technical description of the final version of the web-based platform, latest at the end of the project.

Funded by
the European Union