Botanica Pacifica. A journal of plant science and conservation. 2017. 6(2): 59–68
DOI: 10.17581/bp.2017.06207

METHODOLOGY & APPLICATIONS

An electronic management system for a digital herbarium: development and future prospects

Dmitry E. Kislov *, Vadim A. Bakalin, Elena A. Pimenova, Valentina P. Verkholat & Pavel V. Krestov

Dmitry E. Kislov *
e-mail: kislov@easydan.com

Vadim A. Bakalin
e-mail: vabakalin@gmail.com

Elena A. Pimenova
e-mail: pimenova_garden@mail.ru

Valentina P. Verkholat
e-mail: val.verkholat@yandex.ru

Pavel V. Krestov
e-mail: pavel.krestov@icloud.com

Botanical Garden-Institute FEB RAS
Vladivostok, Russia

* corresponding author

Manuscript received: 29.10.2017
Review completed: 21.11.2017
Accepted for publication: 26.11.2017
Published online: 27.11.2017

ABSTRACT

The paper describes the structure and functional aspects of the electronic herbarium system with a web interface developed at the Botanical Garden-Institute FEB RAS (BGI) in 2016-2017. The main purpose of the system is to provide online access to the herbarium data, including online search operations and the facilities to enter new records into the herbarium database and to generate labels for specimens. The system is therefore multipurpose. It is primarily written in the Python programming language and has several key features: a two step validation process of digitized herbarium records, multi-user and multi-acronym support, semi-automatic herbarium sheet labelling based on entered data, handling of multispecies herbarium records (e.g. cryptogams), flexible taxon-level search and filtering within geographical areas via a web interface or automated search engine relying on HTTP API. The current system is actively used to manage a digital herbarium at the BGI, including its departments in Sakhalin and Amur Branches. The system can be used as well to integrate herbarium information from many other collections.

Keywords: herbarium, data, biodiversity, distribution, collection, digitization, plant

РЕЗЮМЕ

Кислов Д.Е., Бакалин В.А., Пименова Е.А., Верхолат В.П., Крестов П.В. Система управления электронным гербарием: разработка и перспективы развития. В статье рассмотрены структура и функциональные аспекты системы управления электронным гербарием, разработанной в Ботаническом саду-институте ДВО РАН (БСИ) в 2016–2017 гг. Назначение системы – обеспечить онлайн доступ к гербарным данным, включая базовые операции поиска и внесения записей в гербарную базу, а также создание макетов этикеток для гербарных образцов. Таким образом, система управления электронным гербарием представляет собой многоцелевой программный комплекс. Она написана преимущественно на языке программирования Python и обладает следующими возможностями: двухэтапным контролем оцифрованных гербарных образцов, поддержкой одновременной работы нескольких пользователей и управления несколькими гербариями с различными акронимами, полуавтоматической подсистемой этикетирования образцов, а также возможностью введения информации о многовидовых сборах (например, споровых), гибким поиском и фильтрацией его результатов, в том числе по географическим областям, с использованием как web-интерфейса, так и поисковых возможностей на основе HTTP API. Данная система используется для управления электронным гербарием в БСИ, включая его Сахалинский и Амурский филиалы. Система также может использоваться для интеграции гербарной информации коллекций других учреждений.

Ключевые слова: гербарий, данные, биоразнообразие, распространение, коллекция, оцифровка, растение

Today's information technologies are greatly accelerating scientific research by providing easy and flexible ways of accessing data collected from all around the world including herbarium collections. Since the 1990s, many applications for digitization of herbarium have been created. Some provide tools for entering specimen records into databases and publishing them on the Internet. The number of databases is continually increasing: currently with dozens of 'digital herbaria', with all records housed by the Herbarium of the Royal Botanic Gardens Kew (*ca.* 7 million records, *cf.* KEW 1853–2017) and the New York Botanical Garden (ca. 3 million records (NYBG 1891–2017)). Within Russia, the largest digital collection is in the Moscow University (ca. 1 million records (MW 1765–2017)).

While digital herbaria vary in the number and type of stored specimens, their functional possibilities and other characteristics, they all provide tools for searching and accessing stored data via the Internet. These procedures are mostly designed around a web interface, allowing queries via web-forms, or by means of a well-documented *applica-*

tion programming interface (API), or both. Usage of the API is recommended when there is a need to work with the database directly from a statistical/data processing environment (e.g. R, Python, etc.). As an example, the Australian Virtual Herbarium (AVH 2012) has a web interface that provides an opportunity for spatial search requests, as well as batch search requests by selected record codes. Although the AVH has no documented API, it is theoretically possible to simulate HTTP search requests programmatically and then analyze the server response. Such a procedure requires additional tools to investigate peculiarities of client-server communication, since automatic queries, in this case, e.g. from R (R Core Team 2016) or other computational environment, are not supported by the present system.

Many other virtual herbaria have well-developed web interfaces that are capable of storing and retrieving large numbers of specimen records. A brief summary of these is given in the next section (Table 1).

The process of biocollection digitization is still in its infancy in Russia, and only the Herbarium of Moscow State University (MW) has demonstrated a promising start (Seregin 2016).

This paper intends to outline basic functionality and architecture of the web application developed to manage digital herbaria and focuses on finding the best solution for the digital herbarium of the Botanical Garden-Institute of the Far Eastern Branch of the Russian Academy of Sciences (BGI). With the launch of the procedure for entering herbarium specimen metadata into the database, we faced a number of tasks for elaboration of the web application. Among them were: 1) the possibility of simultaneous remote access to the database by many users having different administrative access levels, functions and skills; 2) generating the labels using standardized layout and barcoding; 3) supporting herbaria based on different sampling procedures (e.g. vascular plants, bryophytes and lichens, algae, fungi); 4) supporting and separating data editing processes in more than one herbarium collections having different acronyms; 5) developing a tool for searching by different criteria including synonymic relationships among taxa.

Diversity and structure of herbarium VBGI

The Herbarium at the BGI in Vladivostok (coded by Index Herbariorum as VBGI) was established in 1974. Today, it houses about 150000 specimens. A few specimens stored in VBGI date from the 19th century, but by far the majority of existing records are from the 20th and 21st centuries.

The collection of vascular plants of VBGI includes over 71000 specimens (3500 Lycopodiophyta, Equisetophyta and Pteridophyta, 1500 Gymnospermae and 66000 Angiospermae). The total number of species is approximately 4000 with specimens collected in different regions of the Russian Far East and Eastern Siberia. Overseas collections are represented by specimens from Mongolia, Korea and Japan.

Since 2006, BGI incorporated two geographically remote branches: the Amur Branch located in Blagoveshchensk City and a Sakhalin Branch located in Yuzhno-Sakhalinsk City (respectively AB BGI and SB BGI) with their herbarium collections coded as ABGI and SAKH respectively. The herbarium of the AB BGI contains over 15000, and that of SB BGI more than 30000 specimens of higher vascular plants.

The BGI houses one of the largest collection of cryptogams in Russia. This was established 25 years ago and is now divided into three main branches: collections within Vladivostok metropolis, as well as two other branches. The cryptogam collection in Vladivostok was established in 2000 and devoted mostly to lichens, until 7 years ago, when the bryophyte collections were in large part transferred from the Institute of Biology and Soil Science (VLA) and new purposeful bryological researches were initiated in the BGI. Currently, the approximate number of lichen specimens in the VBGI is 15000, whereas the bryophytes collections exceed 30000 specimens, with over 20000 belonging to the liverworts. The latter is the largest collection of bryophytes in the Russian Far East and, according to available data, in Russian Asia. The recent digitization of cryptogams has resulted in 4000 available online accessions, whereas digitization of lichens is yet to commence.

The cryptogam collection in AB BGI was established at the end of 1990s and consisted of mostly fungi (mainly hyphomycetes and basidial agaricoids), and, to a lesser extent bryophytes (mostly mosses). The approximate number of collections is now close to 10000, but the exact number of specimens is unknown due to as yet incomplete taxonomic revisions of the material. The digitization process is now only beginning with relatively few specimens digitized. The cryptogam collection in the Sakhalin Branch of the BGI was established in 1991 and consisted of mostly lichens. While the digitization process is yet to commence, the estimated number of specimens that need to be digitized is between 5000 and 10000.

It should be mentioned, however, that all branches of the BGI enrich the number of housed specimens by 5–10 % annually due to regular field work and exchange programs (the latter maintained mostly by VBGI).

Herbarium digitization and database management

Once the primary and very general goal was formulated for the development of an online accessible database for herbarium specimens in the BGI herbaria, the following specific needs of the planned electronic herbarium system became evident: 1) providing tools for the entry and editing of database records remotely and simultaneously by multiple users, 2) distinguishing user privileges between regular (non-skilled) and advanced users (e.g. herbarium curators) in generating labels for herbarium specimens. This includes using a standardized layout, supporting and separating data editing processes in two or more herbarium collections belonging to different herbaria having different acronyms, and processing search requests according to prevailing synonymies. An attempt to resolve these and other challenges resulted in the digital herbarium structure outlined below.

Digital herbarium structure

An analysis of the web application's basic features used in the world's most popular digital herbaria and virtual biocollections revealed that well-developed web applications

Table 1. Comparison of the virtual biocollections

Database name	Features			
	Basic search via web-interface	Search within arbitrary selected area	Search API	Specific R or Python library for making search requests
The global biodiversity information facility (GBIF 2001)	+	+	+	+
JSTOR (JSTOR 1995)	+	–	+	+/–
MW: National Digital Bank of Life Systems (MW 1765–2017)	+	–	–	–
The Australian Virtual Herbarium (AVH 2012)	+	+	–	–
Kew Herbarium Catalogue (Kew 1853–2017)	+	–	–	–
Naturallis Biodiversity Center (Naturallis 1984–2017)	+	+	+	+/–
Biodiversity Information Serving Our Nation (BISON 2015)	+	+	+	+
The New York Botanical Garden (NYBG 1891–2017)	+	–	–	–
Botanical Garden-Institute FEB RAS (VBGI 1974–2017)	+	+	+	+/–

Note: +/- denotes that the search API can be exploited via standard (or existed third party) R/Python packages, but there are no official/specific R or Python packages for making search queries to the database

cover all widely used search options and allow automatically generated queries using API (Table 1). The latter could be considered a useful feature as it provides a convenient way to search within any statistical environment that supports scripting.

As currently developed, the electronic herbarium system allows simultaneous data entry by several users. This feature appears to be very important for efficient digitization of large collection and can significantly increase the speed of specimen processing. The multi-user support was implemented on top of the Django web-framework (Django 2017) by using its authentification/authorization component and by providing an additional set of permissions. These permissions define user roles within the digital herbarium system.

The core of the Digital Herbarium Database consists of several tables (currently controlled by the MySQL database server, MySQL 2017), which store all the data on taxon-level names, locations, environmental conditions and other important information for herbarium specimens.

The structure of these tables and their interrelations could be changed during the web application modernization. The current structure of these tables can be found in the <models.py> file at the project's development page Herbarium Management App with Multiuser Support (Kislov 2017b).

The structure of the table, which stores information regarding herbarium specimens, is presented in Table 2. All fields are validated prior being saved as a database record. The record identification dates cannot be set earlier than the actual herbarium collection (collected dates). In addition, it is not possible to set all dates later than the current date (exactly, UTC + 2 days). Such validations render the system more robust to occasional input errors.

One of the main tables in the database stores information about users. Since the web application was developed on top of the Django Web Framework (Django 2017), it employs standard Django's user-model and authentication/authorization subsystem.

Taxon-level names are stored in several hierarchically related tables under <Species>, <Genus> and <Family>. The Species table is connected with the Genus table, and the latter connected with the Family table. Internally, these types of connections are described as foreign key constraints. Genus and Family tables have a simple structure: taxon name, a string field; where the data in this field are internally stored in lowercase format; authorship – genus or family authorship. Prior saving of Genus or Family items is validated according to unique names. These tables were filled in advance with taxon names obtained from theplantlist.org (The Plant List 2017).

The Species table (Table 3) includes fields describing synonymic relationships. These relationships are used by the web application engine to enable searches within species synonyms. The table has a synonym field, which is a self-referential foreign key field. It refers to the current valid species name and is used to describe synonymic relationships among known species. The latter is urgently needed in order to facilitate search requests by old names and to maintain and permanently update nomenclatural changes in modern systematics.

The field 'status' describes assurance regarding the validity of species name. If the species item was added to the database by a regular (not-skilled) user, it has an automatically assigned status 'Recently added'. This means the species needs to be verified by an advanced user. Species items with status 'Approved' or 'From the Plantlist' are treated by the system as trusted. If a herbarium record has a species with non-trusted status ('Recently added') it cannot be published until the referenced species is confirmed as trusted (status 'Approved'). The latter is widely used for cryptogams, where we avoid overloading the total information from the Plant List and organize the manual process of verification of each name to ensure compatibility with the leading checklists and to list synonymy where necessary.

Access rights

To determine access rights to the herbarium database we used a hierarchical model of user roles. A user with the highest privileges heads the top of such hierarchy (**supervisor**). The highest privilege gives universal rights to vary the database records such as editing records in any table,

Table 2. Specimen record structure

Field name	Field Type	Description
ID	**INTEGER**	unique integer number; automatically assigned to each herbarium record when it is saved;
species	**INTEGER**	a foreign key to the table of known species (just a link to the particular species name); this is the only mandatory field;
type_status	**VARCHAR**	available choices are: empty, Holotypus, Isotypus, Paratypus, Lectotypus;
short_note	**VARCHAR**	a note regarding the main species of the herbarium record;
significance	**VARCHAR**	describes the measure of ambiguity regarding species name of the herbarium record; allowed values are <empty> (no ambiguity); aff. (short for Latin <affinis>) or cf. (short for Latin <confertum>);
itemcode	**VARCHAR**	inventory number; can be empty; inventory number should be unique (if provided) within the current herbarium acronym;
fieldid	**VARCHAR**	field number; all alphanumeric characters (char) and some special symbols are permitted;
acronym	**INTEGER**	a foreign key to the table of known herbarium acronyms; acronym is assigned automatically when the herbarium record is saved;
country	**INTEGER**	a foreign key to the table of known countries; the web-application uses ISO-standardized country names;
region	**VARCHAR**	administrative region of the location of the herbarium collection;
district	**VARCHAR**	administrative district of the location of the herbarium collection;
detailed	**VARCHAR**	detailed information about the location of the herbarium collection; assumed that the field stores information about environmental conditions and other important information regarding the collection;
coordinates	**VARCHAR**	geographical coordinates of the location; stored as comma separated values of latitude and longitude given in the WGS-84 reference model;
altitude	**VARCHAR**	altitude; should be given in meters above sea level; internally, it is a string, because a user can provide fuzzy values, e.g. elevations from 100 to 200 meters (100–200 m) etc.;
gpsbased	**BOOLEAN**	boolean parameter, its true value means that the herbarium record position is obtained via Global Navigation Satellite System (e.g. GPS, GLONASS);
collectedby	**VARCHAR**	collectors names separated by commas;
collected_s	**DATE**	the date the herbarium specimen was collected (first day or null if no information provided);
collected_e	**DATE**	the date the herbarium specimen was collected (last day or null);
identifiedby	**VARCHAR**	identifiers names separated by commas;
identified_s	**DATE**	the date the species identification was started (or null);
identified_e	**DATE**	the date the species identification was finished (or null);
devstage	**VARCHAR**	development stage; available values: "Development stage partly", "Life form" or empty string; this field is used at the VBGI to characterize herbarium collections of special type – biomorphological herbarium, and it isn't mandatory for filling;
subdivision	**INTEGER**	a foreign key to the table of known herbarium subdivisions (e.g. bryophyte herbarium, fungi herbarium etc.); auto-filled field;
note	**VARCHAR**	additional information regarding the herbarium collection (everything that wasn't yet included in the previous fields);
created	**DATE**	the date the record was created;
updated	**DATE**	the date the record was updated;
createdby	**INTEGER**	a foreign key to the table of known users; it is assigned automatically when the record is created;
updatedby	**INTEGER**	a foreign key to the table of known users; it is assigned automatically each time the record is saved;
public	**BOOLEAN**	its <true> value denotes that the record is publicly available; records with <false> values are hidden from external access (search via the web-interface or HTTP API);
dethistory	-	reversed foreign key constraint; this field doesn't exist in this table, but there is a table that references to it;
additionals	-	reversed foreign key constraint; this field doesn't exist in this table, but there is a table that references to it;
uhash	**VARCHAR**	reserved.

creating additional users and assigning appropriate rights to work with the database. A built-in limitation of highest privileges nonetheless prevents the deletion of those herbarium records which are already published. Only the highest privilege allows the right to assign acronyms and subdivision of a herbarium record. In other cases these values are assigned automatically depending on relations defined in tables 'Acronyms' and 'Subdivisions'. The structure of these tables is accessible via a project's development page (see the <models.py> file, Herbarium Management App with Multiuser Support, Kislov, 2017b).

A step down in the user privilege hierarchy is the **herbarium (acronym) curator's rights**. Users with acronym curator's rights are eligible to edit herbarium records inside the herbarium with a unique acronym. They cannot edit or even browse unpublished herbarium records attached to another

Table 3. Structure of the species table

Field name	Field Type	Description
ID	**INTEGER**	unique integer ID assigned to each known species when it is saved;
name	**VARCHAR**	species epithet (case insensitive);
authorship	**VARCHAR**	species authorship (case sensitive);
genus	**INTEGER**	a foreign key to the Genus table;
infra_rank	**VARCHAR**	infraspecific rank; allowed values: subsp., var., f., subf., subvar., empty string;
infra_epithet	**VARCHAR**	infraspecific epithet (if the infraspecific rank is not empty);
infra_authorship	**VARCHAR**	the author of infraspecific epithet (if it is given);
status	**VARCHAR**	allowed values: "Approved", "From plantlist", "Recently added", "Deleted";
synonym	**INTEGER**	self-referential foreign key field; it points to the species that is synonymic to the current one (can be null);
updated	**DATE**	updated by the current date each time the species instance is saved.

Table 4. Table of additional species

Field name	Field Type	Description
ID	**INTEGER**	unique integer ID;
herbitem	**INTEGER**	a foreign key to the HerbItem table (Table 1);
identifiedby	**VARCHAR**	species authorship (case sensitive);
identified_s	**DATE**	the additional species was determined (a valid from date);
identified_e	**DATE**	the additional species was revisited (if exists, a valid to date);
species	**INTEGER**	a foreign key to the Species table;
significance	**VARCHAR**	the same meaning as in the Table 1;
note	**VARCHAR**	an arbitrary note for the additional species.

herbarium. Curator's rights of the herbarium subdivision assume those of herbarium curator's rights but only within that herbarium subdivision.

Regular rights enable only a single function, that of database entry. It is possible to edit the records, which were previously created by the same user. However, there is no permission to edit or to browse records created by other users, as well as publish records or to change a species status. Publishing and species name validation (and status changes) are performed only by users with curator's or supervisor's rights.

Typical workflow when compiling aherbarium database includes the following steps: 1) creation of herbarium acronyms, subdivisions, user assignment to appropriate access levels; 2) compiling the database (users having regular rights enter the database with records and new (unknown to the system) species); 3) data validation and publishing (this step is responsibility of privileged users – curators and/or supervisors.

Handling the multispecies specimens, re-identification process and name validation

Each record line in the Table 4 is assigned to the corresponding herbarium record (via the 'herbitem' field) and stores information about additional species. There is no restriction to the number of additional specimens attached to a herbarium record. It is worth noting that the fields 'identified_s' and 'identified_e' have special meaning here. Using these fields it is also a simple matter to trace the identification history of any particular additional species refinements. If one considers a circumstance where a herbarium record is denoted as 'A' with an attached additional specimen denoted as 'B': If the additional specimen was revisited (e.g. in 1 Jan, 2017) and re-identified, one can create an another record (called 'C'). The 'C' record exists along with 'B', but refers to another (refined) species name. In this way, the 'C' record would have 'identified_s' property set to '1 Jan, 2017' and the 'B' record would have 'indentified_e' = '1 Jan, 2017'. The consequence is that the 'B' record is not valid if the current date is later than 1 Jan, 2017, but the 'C' record becomes valid for that date. These records exist in the same time frame and make possible the description of specimen composition for any particular time depending on its refinements. Moreover, taking into account the same plant, the specimen may be re-identified several times. There is no guarantee that the true name is the last, but not the former or not being suggested in the future, the search requests thus deliver information on both the most recent as well as previous identifications .

The species field defined in Table 2 is mandatory for herbarium records of all types (multi- or single species records). If the herbarium record has attached additional specimens, the specimen defined in this field is treated as the main one, and other additional specimens are treated as associates of the second value level.

Two-step validation is another feature of the developed web application. The first step assumes non-skilled operation – entering the data from existing specimen labels into the database (we don't use any OCR-based systems, which are sometimes used as a tool to increase the speed of data entering. Using OCR software is very problematic in our case: almost all existed herbarium labels are handwritten, so it is not easy to recognize text, sometimes even by experienced staff). The next step, performed by advanced users (curators), includes validation of the data, with corrections if needed, and publication of verified herbarium records.

As is usual in plant taxonomy, there are many contemporaneous taxonomic concepts. This results in the appearance of different taxon names for the same (or almost the same) objects in the same dataset. To solve this problem a subsystem of handling species' synonyms was integrated into the web application where every species known to the system could be provided with its synonym(s). This information, in turn, could be optionally used when performing searches over the database records via a web interface or HTTP API.

Herbarium scanning

Herbarium sheet scanning in our system is designed according to a completely independent process. It results in generating an image file with .tiff extension, named by ID (Table 2) or "acronym name + ID" (e.g. VBGI1321) of the herbarium record being scanned. If several image files correspond to a particular herbarium record they are named in the following format using underscore symbol _ VBGI1234_1.tiff, VBGI1234_2.tiff etc.

There are no specific regulations regarding image resolution in herbaria worldwide. In our case we follow to the recommendations from Seregin (2016) to use a resolution of 300 dpi for regular herbarium sheets and 600 dpi for those of highest importance – exemplary specimens or type specimens.

Herbarium records and corresponding images are connected via their unique identification numbers – IDs. Where they exist, corresponding images automatically appear at the bottom of the herbarium record web-pages.

Labels generation

Supplying a specimen with a label and its barcoding is an important step prior to depositing into the herbarium. The label should include important information about the origin of the specimen; environmental conditions where it was collected, collection dates, collector and identifier names etc. To ensure the label creation process is simplified, we developed a subsystem that solves the problem of label generation. Integrated into administrative web interface this subsystem allows the creation of labels as *.pdf files which have a unified layout and platform-independent view (due to specificity of the Adobe Acrobat pdf file format). A sample herbarium label is illustrated in Fig. 1. The user can generate a number of herbarium labels at the same time. All labels are optimally placed onto an A4 page, have the same layout and streamed via http-protocol as a pdf-document (no pdf-files are stored at the server side).

Each label is supplied with a QR-code (we use version 2 of the qr-code generating algorithm with "medium" error correction option) that encodes permanent URI of the current herbarium sheet. Working with the herbarium sheet one can decode the qr-link via a mobile device and obtain the full available information regarding the collection. Along with a QR-code, each herbarium sheet is supplied with a barcode. The later encodes a string – an exact unique identifier of the specimen, a herbarium acronym followed by the specimen inventory number assigned by the system automatically. A sample barcode is given in Fig. 2.

All fields placed on the herbarium label except the 'Place' field, are automatically transliterated to Latin letters if their content is given in Cyrillic. Because the 'Place' field is not transliterated, we recommend this field should be entered directly in English.

For multispecies herbarium collections, common, for example, in fungi and bryophyte herbaria, the system can generate labels of another type – an envelope label. Such types of labels are designed to be printed on the paper of A4 format presuming that the latter will be folded into an envelope containing the specimens (Fig. 2).

Herbaria based on different sampling protocols

Thus developed, the web applicationcan handle two general types of herbarium records: single- and multispecies. Single species herbarium records are the norm when dealing with vascular plant collections. They assume that each herbarium sheet corresponds to a single plant species.

Multispecies herbarium records are usual for cryptogam collections, e.g. bryophytes and fungi. In this case, each herbarium record can provide information regarding several species included in the collection.

The web application supports both types of herbarium records. The support of multispecies records is realized by assigning to a user an appropriate set of rights (available for all cryptogams excluding vascular cryptogams). When a user is marked as an "editor of multispecies collections", then that user has an additional panel at the bottom of the edit web-page of the herbarium record. This panel allows the entry of information about additional species that belong to the collection. All data entered in this panel are stored in a special table with the structure described in Table 4.

The support of herbaria with different acronyms

Another feature of the electronic herbarium system is that its branching structure implies definite groups of herbarium specimens. There are two types of such specimen groups. The first is defined by the herbarium acronym. The web application facilitates separate user activities between acronyms: thus it is possible to set permissions in the way that the user could be able to edit and browse only those records belonging to a specific herbarium acronym. Therefore, the web application provides for the organization the multi-acronym digital herbarium storage, where curators are responsible for their sections as well as regular users. Each acronym can be supplied with its own address, abbreviation and logo. These parameters are used when aherbarium sheet label is generated. Herbarium branching

opens the opportunity to accommodate a large number of differently organized herbaria in the same system and to provide unlimited access to the herbarium data located even in isolated remote herbaria.

The herbarium branching concept extends beyond that of separating plant specimens by their connection to different herbaria in spatial terms. Another type of herbarium branching is the specifying of herbarium subdivisions. This type of specimen grouping is useful within herbarium acronyms when it is needed to distinguish different plant group collections, e.g. bryophytes, fungi or vascular plants. The web application system provides curators with the number of rights allocated to the sectors for which they are responsible.

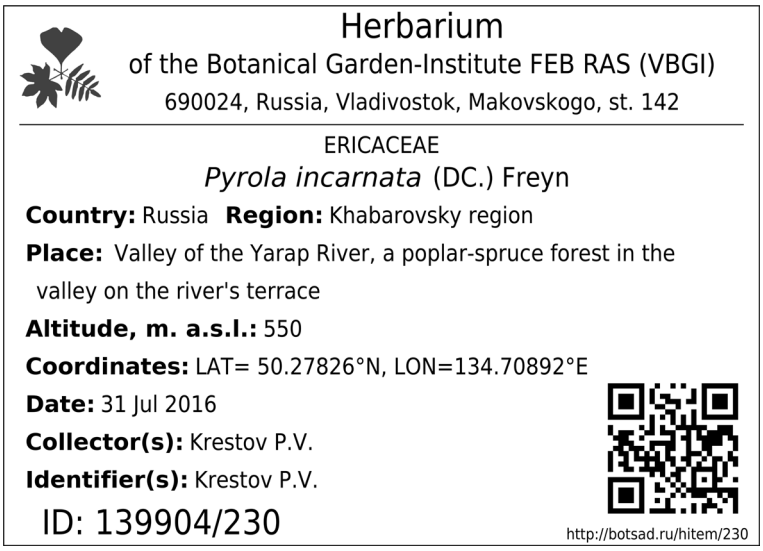

Figure 1 Sample herbarium label

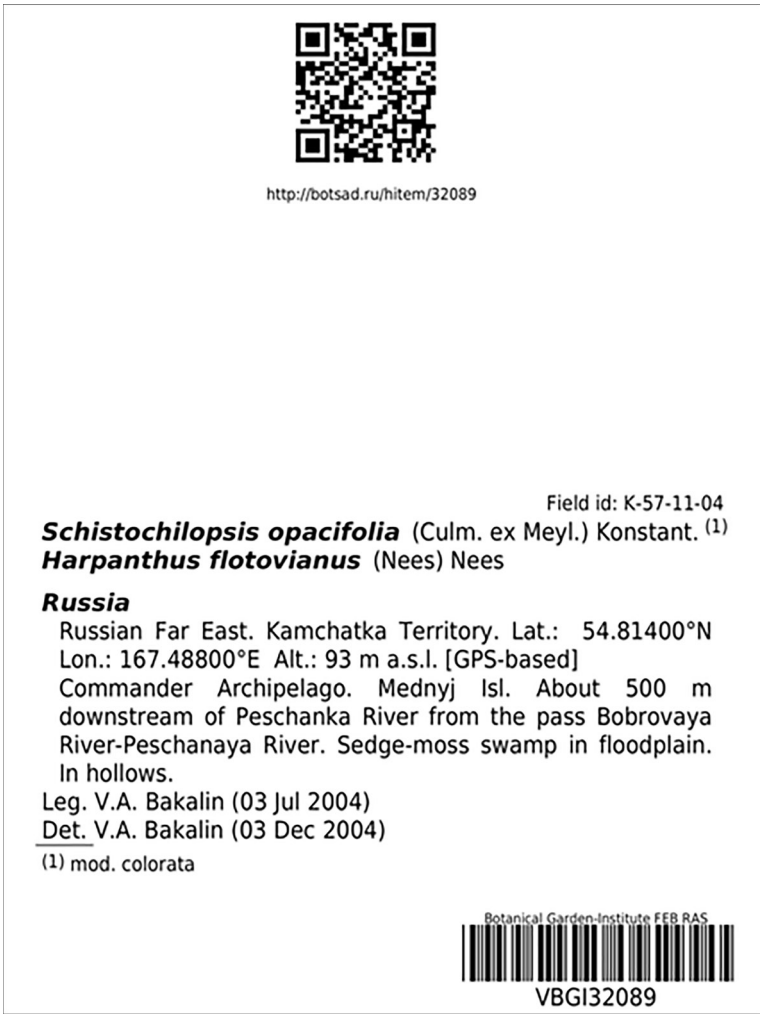

Figure 2 A fragment of the envelope label of the multispecies herbarium collection (full size envelope label is of A4 paper format)

Accessing the data

There are three regular ways of data access in a digital herbarium system, namely using the web administration interface, the publicly available web search interface or the HTTP API engine. The web administration interface is primarily focused on data entry and editing processes, and does not provide flexible search tools. This method of data access is available to those users with rights to make changes in the database.

The web search interface (currently available through the link http://botsad.ru/herbarium, Fig. 3) allows users to search by family and/or genus names, species epithet, collection and identification dates or selected area on the World map. Only published records will be shown as search results. The only restriction, when searching via the web search interface, is that the current implementation is restricted to only 'AND'-type search request at a time. To combine results of two different search queries, i.e. perform 'OR'-type search request, we recommend HTTP API.

The application programming interface and its supplement to the digital herbarium in particular, is a modern feature that facilitates interaction with the database in a way that is familiar to most developers and data scientists. Requirements for processing large amounts of data originating from different sources can thus become a routine procedure in contemporary researches. API could be considered as a well-documented set of commands that are sent to the system. As a result, API returns a response according to the stated format. The developed web application supports HTTP API that is focused on performing search queries. The HTTP API service is also designed to be ReST compliant (REST API 2017), self-explanatory and transparent from the viewpoint of search query construction. A response returned by the web application is a JSON-string format. A detailed explanation of all parameters applicable to request/response HTTP-messages along with examples of using the HTTP API service can be found in Digital Herbarium's HTTP API Description (Kislov 2017a).

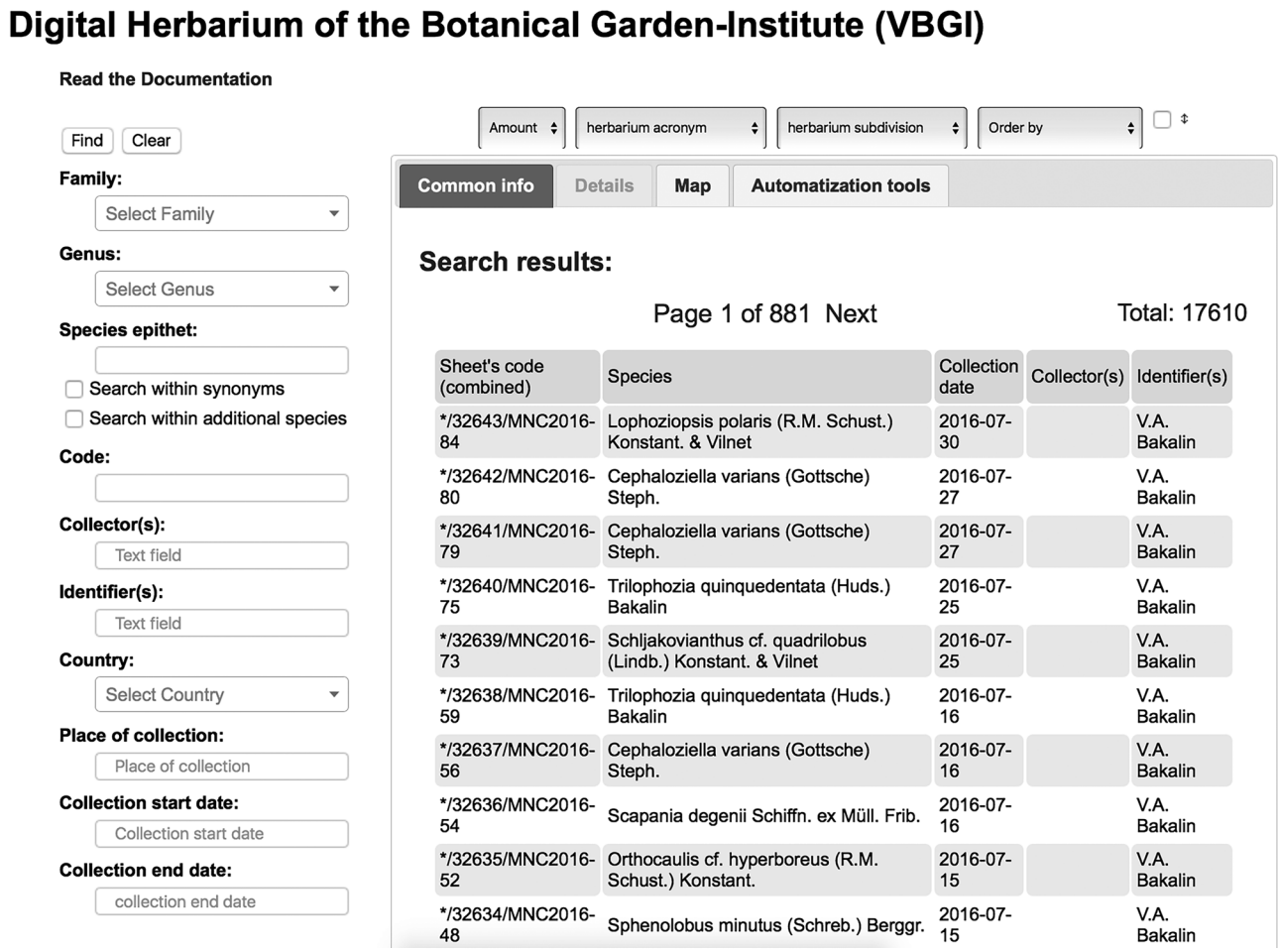

Figure 3 Web interface of the digital herbarium (as of state on 27 Nov 2017)

Nowadays, many digital collections around the world provide flexible interfaces that enable basic operations with records using programming languages. For example, the Global Biodiversity Information Facility (GBIF 2001) provides a well-documented API for search queries. In GBIF, search facilities are publicly available to all users. At the same time, record changing/creation operations, provided as PUT, DELETE and POST HTTP-methods, require authentication. GBIF is not the only digital resource providing API as many others possess the same opportunities. Among them are JSTOR (JSTOR 1995), Naturalis (Naturallis Biodiversity Center: BioPortal, Naturallis 1984–2017), and BISON (BISON 2015). The main reason why increasing numbers of digital databases provide publicly available API, is that this type of data access provides ample opportunities for undertaking a wide range of studies. If one uses R (R core team 2017) or another computational (and/or statistical) software to process the data, it would be more convenient access some portion of external data from that computational environment directly, rather than through its web interface.

HTTP API provides the opportunity to work behind the web interface of our digital herbarium database. This API is focused on processing search requests only as no record changing operations are permitted. The full set of allowable parameters that can be passed to the search engine, as well the structure of JSON-formatted responses from the server, are described in Digital Herbarium's HTTP API Description (Kislov 2017a).

To illustrate the utility of HTTP API, we consider here a hypothetical task as a part of the problem of species distribution modeling (SDM). To build a model one needs to obtain geographical coordinates of all records for a particular species. There are two ways to obtain the data from the database: either via its web interface or using an HTTP API service. Interacting with the web interface assumes some manual operations – such as building queries via the web interface and saving the data from each page of the search results. Our digital herbarium's web-search engine lists only 200 records per page. Therefore, if the search results in exceedingly high data lines, it may be a problem to obtain all of them using this approach.

Another way consists of exploiting the HTTP API service and enabling search queries directly from the computational environment which is used to build the SDM. If one uses R programming language, it will require just a few lines of code to extract, for example, all records belonging to a particular genus.

As an example, we consider the task of obtaining all herbarium records belonging to the *Rhododendron* genus. All that is needed for such a search request using HTTP API is a preinstalled "jsonlite" R-package (Jeroen et al. 2017) and a

few lines of R code. Because responses of the HTTP API service are json-formatted strings, the user is free to choose other an R-tool for automatic parsing json-strings, although "jsonlite" is to be preferred according to results from our tests with the system. For example:

```
library(jsonlite)
data<-fromJSON('http://botsad.ru/hitem/json/?genus=Rhododendron')
data$data
```

If an "rgdal" library is preinstalled, it is easy to filter search results by any particular area, e.g. retrieve all herbarium records within a contour defined in an ESRI shape-file.

To make a commitment to follow the best practice in constructing search queries we recommend refering to the official documentation of the HTTP API service (Kislov 2017a) and to explore the links concerning examples in Python and R programming languages.

Entering the data: routine workflow

Typical workflow of data entry into the database includes (Fig. 4): 1) user authorization; 2) entering the specimen's metadata; 3) verifying the correctness of the entered data (e.g. species names, and speciemn's metadata); 4) publishing the record.

The web application includes a database of known species names (see Table 3). If a herbarium collection represents a new (unknown to the web application) species, a user can enter new data regarding such species (species epithet, authorship, etc.), but they will be marked as a recently added (untrusted) species items and will require additional validation that should be performed by a skilled user, e.g. a user with curator's rights. Species validation assumes setting a flag of species' status (see the Status field in Table 3) to 'Approved'. It is not possible to publish a herbarium record of an unapproved (untrusted) species.

Under these provisions, the last two steps at the diagram (Fig. 4) should be completed by a skilled user. Hence, users who are able to approve taxon-level names and publish specimen records must have curator's privileges. If the record was published, it is treated as trusted, and in this way it is possible to generate labels for such herbarium records.

Further developments

As described above, the integration of biocollections into a common informational system would be very efficient today. At the same time, the development of virtual biocollections is still at the initial stage within Russia. As of 2017, less than ten Russian digital herbaria are in operation and are accessible via the Internet (not all of specimens preserved in those herbaria are available online). The most famous and largest of these is the digital collection of the Moscow University (Seregin 2016).

The digital biocollection development is a very important step in providing the worldwide scientific community with comprehensive data access. It will enhance possibilities for improved scientific collaboration between Russian researchers and their colleagues abroad. That, in turn, would facilitate broad-scale studies involving large data sets of species occurrence and herbarium collections data. After the initial input of all label data we plan to improve procedures

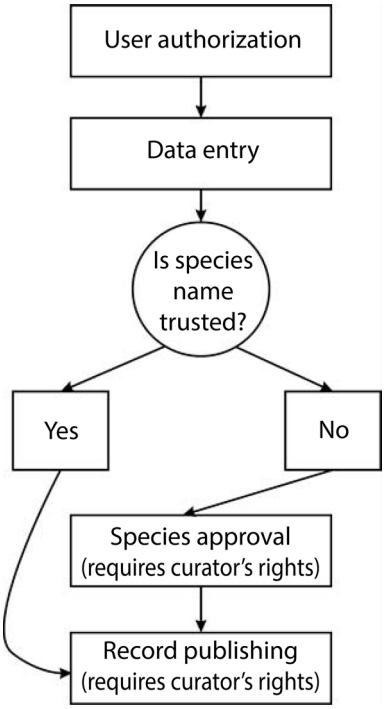

Figure 4 Data entering workflow: basic steps

for generating available online labels with exact geographic coordinates using e.g. Google Earth geo-information system. Until now, for example, the only cryptogam specimen input to the database is accompanied with a manual search of geographical coordinates (where they are not recorded on the label) of geographic locations mentioned in the label. This work leads to decreasing data 'input productivity' because of many names are difficult to locate in available maps. Nevertheless, this procedure is urgently needed because it provides more efficient searching of specimen data using automated requests.

A consolidated system for storing and retrieving digitized herbarium data should be developed taking into account basic requirements that are already accepted by the worldwide scientific community. Among these requirements are the possibilities to obtain stored data using web interface, API and filtering spatially distributed data by specific/user defined areas.

The electronic herbarium system developed at the BGI satisfies all basic requirements and provides ready access to the data using a web interface or API. In the meantime, the digital herbarium of the BGI is a developing project where, as far as the developed web application is concerned, there remain many improvements that still await its full implementation.

One of them is the migration from MySQL database management system to PorstgreSQL (PorstgreSQL 2017) with the PostGIS extension (PostGIS 2017). This would allow geographical (e.g., searching in polygonal areas) search requests processed at the database level. Today, at the database level it is possible only to filter search by rectangular areas (according to latitude or longitude lines) when performing a geographical search. Nonetheless, one can still

use a two-step filtering process: 1) locating all records belonging to the bounding box of the selected area and 2) filtering using programming language (e.g. R or Python). However, the second step is performed outside the database engine and may be inefficient especially if the database stores large amounts of specimen data.

Another improvement implies the development of a subsystem that facilitates online measurement of herbarium sheets directly on its images. The calibration palette (a set of colored squares) placed on each herbarium sheet before scanning could be used for color adjustment purposes, but also to automatically compute scaling factors between pixels and systematized units of length (e.g. centimeters).

Currently, the project's roadmap assumes migration to a PostgreSQL database engine and the development of flexible engine handling for complicated search queries (including full support of AND-, OR- and NOT Boolean operations).

ACKNOWLEDGEMENTS

The work was supported by the Federal Agency of Scientific Organizations (special program on support of biocollections to BGI FEB RAS, AAAA-A17-117073110007-8) and partly by the Russian Foundation for Basic Research (grant 17-04-00778 to Vadim Bakalin). Authors are deeply indebted to our collegues, who worked with specimen labels when filling database, for constructive discussions in the course of electronic herbarium system development and for testing the system: Natalia Gorodilova, Ksenia Klimova, Kirill Korznikov, Ekaterina Petrunenko, Tatiana Stupnikova and Svetlana Yurchenko. We thank Prof. Andrew N. Gillison for the linguistic editing of the manuscript.

LITERATURE CITED

AVH 2012. *The Australasian Virtual Herbarium*. Available at https://avh.chah.org.au/. Last accessed: 31 October 2017.

BISON 2015. *BISON: U.S. Geological Survey. Species occurrence data for the Nation—USGS Biodiversity Information Serving Our Nation.* U.S. Geological Survey General Information Product 160, 1 p. http://dx.doi.org/10.3133/gip160. Available at https://bison.usgs.gov/#home. Last accessed: 31 October 2017.

Django 2017. *Django: The web framework for perfectionists with deadlines*. Available at https://www.djangoproject.com/. Last accessed: 31 October 2017.

GBIF 2001. *The Global Biodiversity Information Facility. What is GBIF?* Available at http://www.gbif.org/what-is-gbif. Last accessed: 31 October 2017.

Jeroen, O., D.T. Lang & L. Hilaiel 2017. *A robust, high performance JSON parser and generator for* R. Available at https://cran.r-project.org/web/packages/jsonlite/jsonlite.pdf. Last accessed: 31 October 2017.

JSTOR 1995. *JSTOR Global Plants*. Available at https://plants.jstor.org/. Last accessed: 31 October 2017.

KEW 1853–2017. *KEW: Herbarium Catalogue*. Available at http://apps.kew.org/herbcat/gotoSearchPage.do. Last accessed: 31 October 2017.

Kislov, D.E. 2017a. *Digital Herbarium's HTTP API Description*. Available at http://botsad.ru/herbarium/docs/en/http_api.html. Last accessed: 31 October 2017.

Kislov, D.E. 2017b. *Herbarium Management App with Multiuser Support*. 2017. Available at https://github.com/VBGI/herbs. Last accessed: 31 October 2017.

MW 1765–2017. *MW: National Depository Bank of Live Systems. Moscow Digital Herbarium*. Available at https://plant.depo.msu.ru/. Last accessed: 31 October 2017.

MySQL 2017. *MySQL. The world's most popular open source database*. Available at https://www.mysql.com/. Last accessed: 31 October 2017.

Naturallis 1984–2017. *Naturallis Biodiversity Center: BioPortal*. Available at http://bioportal.naturalis.nl. Last accessed: 31 October 2017.

NYBG 1891–2017. *The New York Botanical Garden. International Plant Science Center*. Available at http://sciweb.nybg.org/science2/hcol/allvasc/index.asp. Last accessed: 31 October 2017.

PostGIS 2017. *Spatial and geographic object for PostgreSQL*. Available at http://postgis.net/. Last accessed: 31 October 2017.

PostgreSQL 2017. *PostgreSQL*. Available at https://www.postgresql.org/. Last accessed: 31 October 2017.

Python Software Foundation 2001. *Python Language Reference*. Available at http://www.python.org. Last accessed: 31 October 2017.

R Core Team 2016. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna. Available at https://www.R-project.org/. Last accessed: 31 October 2017.

REST API 2016. *What is REST API? RESTful API Tutorial*. Retrieved 29 September 2016. Available at https://restfulapi.net/. Last accessed: 31 October 2017.

Seregin, A.P. 2016. Making the Russian flora visible: fast digitisation of the Moscow University Herbarium (MW) in 2015. *Taxon* 65(1):203–209.

The Plant List 2017. *The Plant List*. Available at http://www.theplantlist.org/. Last accessed: 31 October 2017.

BGI FEB RAS 1974–2017. *VBGI: Herbarium of the Botanical Garden Insitute FEB RAS*. Available at http://botsad.ru/herbarium/. Last accessed: 31 October 2017.