Contents

1 Aspirational behaviors for the SKOL web site

The Synoptic Key of Life is an effort to understand all the formal taxonomic descriptions in the biological literature with machine learning and make them accessible to the amateur taxonomist.

The core of the SKOL web site is a search capability which allows an aspiring taxonomist to enter a description of a specimen and find the most relevant literature.

Key: | Token | Meaning | |——-|———| | **D** | Done as of 2026-01-15 | | **M** | Miniumum Viable Product | | **I** | Planned for IST690 | | **S** | Stretch goal for IST690 | | **N** | Nice to have | | **R** | Post IST690 research project |

1.1 Project principle

1. All data is Open Access. **D**
2. All source is Open Source (GPLv3). **D**
3. All essential components are Open Source. **D**

Note: neo4j is not open source, but is not an essential component. It is only used for manual data validation.

1.2 Capabilities

1.2.1 Navigation

The following buttons should be available on every page: 1. Home takes you to the search landing page. **I** 2. Account takes you to an account management page. **I** 3. Report issue with this page. This allows web site bug reporting. **I** 4. A hamburger menu with: **I** * Contact Admin. This takes you to a contact page for the admin mailing list. **I** * Other search capabilities are reachable from this menu. **S**

1.2.2 Ingestion (scraping)

1. SKOL honors robots.txt. **MD**
2. SKOL honors HTTP 429 Too Many Requests by backing off the requested amount. It has an exponential backoff algorithm if the necessary headers are missing from the 429. **MD**
3. Each SKOL ingester stoplists a web site after it gives the first HTTP 403 Permission Denied. This is only for the duration of the ingester session, so that if permission error are resolved retries are automatic. **D**

1.2.3 Description management

1. Look up the n closest descriptions in the literature to the description in progress. **MD**
2. Record each Taxonomist's descriptions. Make them searchable and editable by the owner. There are explicit save and cancel buttons. **I**
3. Associate metadata with each description: MO link, iNat identifer, GenBank reference(s). **I**
4. Taxonomist descriptions become part of the searchable literature. **S**
5. The Taxonomist may record their identification of a description. **MI**
6. Matches are each provided with a button that allows the Nomenclature to be copied to the identification field. This will also add an Identifier record with the login ID of the Identifier. **N**

1.2.4 JSON description format

1. Consider providing a JSON form for the description. This could be either a separate alternative to the text description, or a format that could be generated on the fly from the taxonomist description. **R**

1.2.5 Account management

1. Let people create new accounts by providing an email address. **DI**
2. Allow password resets via email. **DI**
3. Allow password resets via text message. **N**
4. Accounts can be Taxonomist, Identifier, or Admin (do we need Moderator?). Identifications made by Taxonomists are not publicly visible unless they are the owner of the description. All identifications by Identifier or Admin are publicly visible. Admins can view, edit, or remove any data. **S**
5. A Taxonomist account is upgraded to Identifier after a set number of their identifications are coroborated by Identifiers. **N**
6. All users can set their notification preferences. **N**
7. Something about reputation scores driven by corroborated identifications? **R**

1.2.6 Menu description aid

1. Provide a set of cascading menus where each level of the menu is a feature, subfeature, optional subsubfeature etc, and finally values. **I**
2. The features and subfeatures are pulled from the JSON representations of descriptions generated by our SLM. **I**
3. The cascading menus are implemented in the style of crucial.com's RAM search tool. The primary feature menu would populate a subfeature menu which can populate a subsubfeature menu, etc. The leaf menu contains values. **I**

1.2.7 Search capabilities

1. The search updates automatically after a short period. This behavior can be disabled or slowed if load gets too high or at the discrtion of an Admin. **I**
2. The Taxonomist can search all Nomenclature fields. **S**
3. The Taxonomist can limit their description searches to their own descriptions. **N**
4. An Identifier or Admin may limit their description searches to an arbitrary user. **N**
5. Taxonomists may subscribe to get notifications for new descriptions. (Is there a sensible or scalable way to limit scope of the reported descriptions? TODO) **IR**

1.2.8 Feature selection tool

1. A classification decision tree built automatically from the top n matches is displayed to the Taxonomist. This tree guides the choice of features to look at. **SR**
2. The decision tree is clickable to update the description. **S**

1.2.9 REST Interface

1. Every behavior that is accessible from the UI is accessible via REST. Automation using the site is encouraged. **MI**
2. Something something REST authentication and tokens... TODO **R**

1.2.10 Permissions

1. Taxonomist's descriptions are universily visible unless marked as private. **S**
2. The private marker allows a Taxonomist a one year embargo to publish before revealing their data. After 1 year, descriptions are made public. Admins may view all descriptions. **S**
3. It is possible to search for just the descriptions of a particular Taxonomist. **S**
4. Descriptions may be deleted by the creating Taxonomist or an Admin. **S**
5. Any authenticated user can flag a description as problematic. **I**

1.3 Site management

1.3.1 Monitoring

1. New accounts are reported to a mailing list and logged. **I**
2. There is a daily report of descriptions or identifications containing URLs, mailed to the admin mailing list. This is to detect spammers. **I**
3. Storage growth is monitored. (Ideally, we'd have a forcast of storage exhaustion. Do we need a pariodic report?) **R**
4. Additional resource monitoring (CPU, GPU, RAM, bandwidth) is available. (Initially, standard Linux tools should suffice, but reporting and alerting may be necessary.) **N**
5. Do we need to detect sock puppets? The theory is that Identifiers are more trustworthy than general Taxonomists. **R**

1.3.2 Backups and mirroring

1. All user data is backed up locally and offsite–account information, descriptions, identifications. **I**

2. All redis data is mirrored to a second host for availability. The important redis data records are easily rebuilt (less than 1 hour). **S**
3. All couchdb data is mirrored to a second host for availability. Non-user data in couchdb originally comes from other web sites and can theoretically be recovered off the web. **S**
4. All couchdb data is backed up locally. **I**
5. Restore the web site from 1 and 4. **I**
6. All source code is in github.com. **MD**

1.3.3 Security

1. Production secret credentials are always encrypted on media. Getting gpg-agent to interoperate with cron is possible but pretty elaborate. **R**
2. Piggy's password failsafe documents the key passwords. Piggy's wife Eve knows how to access this. **I**

1.3.4 Email configuration

1. The postfix MTA uses Cyrus SASL for secure SMTP. https://www.postfix.org/SASL_README.html **N**
2. All DNS modern DNS records set up for synoptickeyof.life. **S**