Potential Project Proposal

Taxa Tree Generator

Max Patiiuk

May 1st, 2023

Sponsor: Specify Collections Consortium <support@specifysoftware.org>

Table of Contents

Introduction

Justification for the Project

Specify 7 is the most popular and capable scientific collection management software. It is used for managing preserved specimen collections. Besides biology, it is used in the geology field. One of the most common needs for new institutions migrating to Specify 7 is to upload a tree of life from a given authority into their database so that it can be referenced during operations on the data or when importing data.

Tree of life data (also known as taxonomy) is managed by several taxonomic authorities. A given institution has its preferred source of authority data. The most common sources include GBIF, Catalogue of Life, ITIS, and WoRMS.

Besides getting the names and ranks of tree nodes, institutions also want the associated metadata like common name, authorship, citations, vernacular names, status (extinct, living), acceptance level (accepted, provisionally accepted, synonym), and links back to the authority's webpage.

Additionally, since the tree of life for some disciplines consists of millions of species (notable Entomology), there is a need for institutions to be able to cherry-pick specific branches from the tree, the branches that their collection is primarily focused on. This reduces the size of the database, thus improving the performance of updates, searches, and backups.

Unfortunately, at the moment there is no single tool that can accomplish all of the above requirements while supporting several data aggregators. Existing solutions commonly involve a lot of manual labor, which, besides being expensive, is error-prone - and discovering a mistake in a spreadsheet of millions of rows is particularly tricky.

Thus, there is a need for a tool that can satisfy these goals. This will simplify the digitization process for many biological collections and improve the quality of data.

Main Project Deliverables

Back-end:

- Mechanism for downloading data from several authorities and converting them to a common format that can then be used to generate the final output
- Automated update of cached taxonomy data from authorities to make sure generated files always use up-to-date data
- Ability to produce final output as a TSV, XLSX, or SQL file
- Unit tests to ensure the correctness of the implementation
- Infrastructure that scales easily to facilitate high and irregular demand

Front-end:

- Ability to select particular branches of the taxa tree for export
- Ability to select which taxonomic ranks and subranks to include in the export
- Ability to select which metadata should be included (i.e, citations, vernacular names)
- User Interface tests to ensure the correctness of behavior

Required Team Skills

The team must be competent with the following technologies:

- JavaScript (for back-end and front-end)
    - Justification:
        - This is the best option for creating a web interface in
        - At the same time, using the same language on the back-end will allow for code reuse.
- React (for user interface)
    - Justification:

        - This is the most popular interface-building framework that encourages code reuse and functional programming.
- TypeScript (for back-end and front-end)
    - Justification:
        - This is an extension of JavaScript that adds strict type checking, resulting in much stronger static checking, and thus better developer experience and fewer bugs. Also, it's becoming the industry standard and it will soon be uncommon to find modern web project not using TypeScript
- Next.js (for back-end and front-end)
    - Justification:
        - This is the most popular end-to-end web framework. It handles both the back-end and front-end logic of a website, allowing for greater code reuse. It also comes with server-side rendering and progressive rendering out of the box.
- Jest (for unit tests)
    - Justification:
        - This is the most popular solution for writing unit tests in JavaScript. Given the size of some generated taxa trees, manual testing will be all but impossible, so automated testing using Jest will be essential
- Docker (for DevOps)
    - Justification:
        - Developing and deploying software in containers using Docker is becoming a big DevOps trend. Each service of an application (web server, back-end, static files, database) could be housed in a separate container. Each container has limited and clearly defined responsibilities.

Docker allows for infrastructure-as-code. All of this allows for a much more manageable and scalable infrastructure.

Main Team Tasks/Roles

- Back-end
- Front-end
- DevOps
- Testing
    - Writing automated tests
    - Quality assurance
    - Requirements compliance assurance
- Documentation and user webinars
- Ongoing maintenance and updates
- Management of the project deliverables, priorities, and deadlines

My Desired Role and Why

I would like to play the role of a front-end and back-end developer (i.e, full-stack developer) who is also available to advise on different technical aspects of the project, provide leadership when needed, and facilitate training and onboarding for team members.

I would like this role because this goes well with my passion for creating something/writing some cool project into existence while giving me the opportunity to provide guidance on challenging aspects of the project as per my experience and give back to the industry by training and mentoring the next generation of engineers.

Project Sales Pitch

Digitizing the world's biological data is of great importance to the scientific community. It empowers research into climate change, diseases, evolution, and many other crucial subjects.

At the same time, the biological collections field is severely underfunded and lacks sufficient software to fully automate all laborious aspects of cataloging specimens and managing a collection.

This project aims to make it easier to jump-start a new collection in Specify 7 (leading collection management software), by creating a great tool for sourcing and reshaping taxonomic data to fit the collection's needs.

Project Plans

Summary

Plan name	Summary	Justification
Project Initialization		
Assumption log	Lists assumptions made regarding demand, technical capabilities, and resources (including human resources)	Allows to keep track of what things we assume are given to us in order for the project to succeed. The assumption log goes well together with the risk management plan
Communication management plan	Outlines communication preferences and methods on the team	Ensures effective and efficient collaboration while reducing conflicts and miscommunication
Quality management plan	Outlines the process for ensuring high quality of implementation and compliance with requirements	Formalizes the process of quality assurance, thus increasing the quality
Milestone list	Gives a high-level overview of stages of development of the project	Gives structure to the process and a reason for celebration once the milestone is reached
Stakeholder management plan +	Lists stakeholders and how project	Provides a way to prioritize

register	members will interact with them	stakeholders and effectively communicate with them
Risk management plan	Outlines potentials issues and roadblocks as well as strategies for mitigating them	Encourages proactively work and plan on reducing negative impact of adverse situations
Project analytics		
Cost estimate	Summarises expected and unexpected expenses	Informs the size of the requested grant for the implementation of the project
Schedule draft	A rough outline of how project development may play out.	The schedule draft does not represent the actual way it will play out but serves as a way to get an overview of what work lies ahead and insights into possible problems.
Quality metrics	Quantifiable metrics into the state of the application, codebase and team performance	Gives an on-the-glance insights into the project
Gantt chart	Visualizes the estimated time needed to complete individual requirements and the relationships between them	Provides a great way to estimate the completion date, to review potential bottlenecks and the relationships between project tasks
Work breakdown structure	A document outlining required tasks and subtasks	Gives idea of what needs to be accomplished on the project
Project progress management		
Burndown chart	Keeps track of completed items, remaining items, and current trajectory	Provides a clear and simple way to see the development velocity and predicts the need for the change of deadline/strategic reduction of scope
Change request document	A formal way to propose, request and review a change to the core requirement list of the project	A structured way to make informed decisions on whether a requirement should be changed
Lessons learned register	A central database for the insights and lessons learned in the process of developing this or previous projects	Allows every team member to benefit from lessons learned by any team member. Reduces repetition of mistakes. Provides a chance for reflection
Issue log	Keeps track of discovered issues as well as remaining missing features	Gives visibility into remaining tasks and the progress on each
Progress report	Outlines completed tasks, challenges encountered and remaining issues	Keeps stakeholders informed of completed and remaining work

RACI chart	Outlines high-level roles on the project and interactions between roles	Clearly outlines the task and responsibilities of each team member
Scope management plan	Outlines deliverables and how completion of deliverables will be verified	Provides a clear definition of what constitutes a completed deliverable
Team contract	An agreement between team members about code of conduct, communication and other behaviors	Reduces team conflict, allows to set a standard of behavior and provides an opportunity to resolve differences early on in the project
Team performance assessment	Formal evaluation of performance of individual contributor	Acts as a basis for promotion and termination decisions

The following are templates of a subset of above mentioned project management plans

Project Initialization

Assumption log

ID	Assumption Description	Category	Owner	Due Date	Status	Actions

Communication management plan

Objectives

Stakeholders

Stakeholder Name	Role	Responsibilities

Communication Requirements

Stakeholder Name	Frequency	Method	Content

Communication Schedule

Communication Activity	Audience	Date	Method

Communication Plan Management

Milestone list

Project Name:

Milestone	Description	Estimated Completion Date

Project progress management

Cost estimate

Project Name:
Date:

	Internal	$/hour	Internal	External	$/hour	External	Total	Non-labor $	Total cost
WBS Categories	Labor		$ Total	Labor		$ Total	Labor		

Project progress management

Change Request document

Date
Project Name:
Date Request Submitted:
Title of Change Request:
Change Order Number:
Submitted by:
Change Category: Scope Schedule Cost Technology Other

Description of change requested

Events that made this change necessary or desirable

Justification for the change/why it is needed/desired to continue/complete the project

Impact of the proposed change on

Scope

Schedule

Cost

Staffing

Risk

Other

Suggested implementation if the change request is approved

Required approvals

Name/Title	Date	Approve/Reject

Issue log

Date:

Issue #	Issue Description	Impact on Project	Date Reported	Reported By	Assigned To	Priority(M/H/L)	Due Date	Status	Comments

Lessons learned register

ID	Date Identified	Owner Name	Category	Situation	Recommendation

Progress Report

Date:
Project Name:
Reporting Period:

Work completed this reporting period

Work to complete next reporting period

What’s going well and why

What’s not going well and why

Suggestions/Issues

Project changes