Database Search Utility Requirments and Technical Notes

Central Coast Climate Collaborative

Dominic Gaiero
dgaiero@calpoly.edu

Revision: 0.1

Contents

Product Delivery Notes

An MVP version of this web application will be available by mid-July 2019.

1 Executive Overview

This web application will serve as a lookup database for institution personal, faculty and collaborators to use for information request. This web application shall reside on a website open to the public and provide a view to allow easy searching of database objects. The first type of user to look at is an institution. An institution may wish to search for another institution offering

2 Development Stack

For a complete list of all expected python (core) dependencies for this project, please reference appendix section A

This web application will utilize the Django web framework. For front-end development, React will be utilized.

3 Security Considerations

- This web application will be open on the Internet to anyone who wishes to access this website. All non-administrative pages of this web application will not be accessed controlled.
- Administrative sections of this web application will be accessed controlled through the built in mechanisms provided via the Django web framework.

4 Production Requirements

4.1 Minimum Server Hardware Requirements

The server hosting this web application should have:

1. 512 MB RAM
2. 1 CPU Core
3. 2 GB free HDD space (allocated size mostly for database.)
    (a) If the database if hosted on another node, then this requirement can be far smaller (500 MB).
4. Appropriate bandwidth
    (a) For a low-usage web application such as the one being built, the bandwidth that is available from the Central Coast Climate Collaborative's website will be appropriate.

4.2 Server Software Requirements

Development of this web application shall be supported on both Nginx and Apache. The database for this web application is supported on both PostgreSQL and MySQL. The choice to support both Apache and MySQL is due to the fact that the preferred deployment of WordPress uses both of these technologies.

> **NOTE:** The preferred method for configuration is on Nginx with PostgreSQL.

4.2.1 Baseline Considerations

- The server, for any configuration will require `sudo` mode privilege escalation to implement the setup for a production environment.
- The server will also require Python3 installed with access to install other python packages (those specified inside of the `requirements.txt`)
- The server that will host this web application will have git version management software installed.

4.2.2 Apache Configuration

- Apache version $\geq$ 2.4 is recommended.
- Apache configuration requires the `mod_wsgi` package to be installed on the Apache server.
- mod_wsgi $\geq$ 2.0 is recommended
- For more information on Django and Apache setup, please reference: `https://docs.djangoproject.com/en/2.2/howto/deployment/wsgi/modwsgi/`

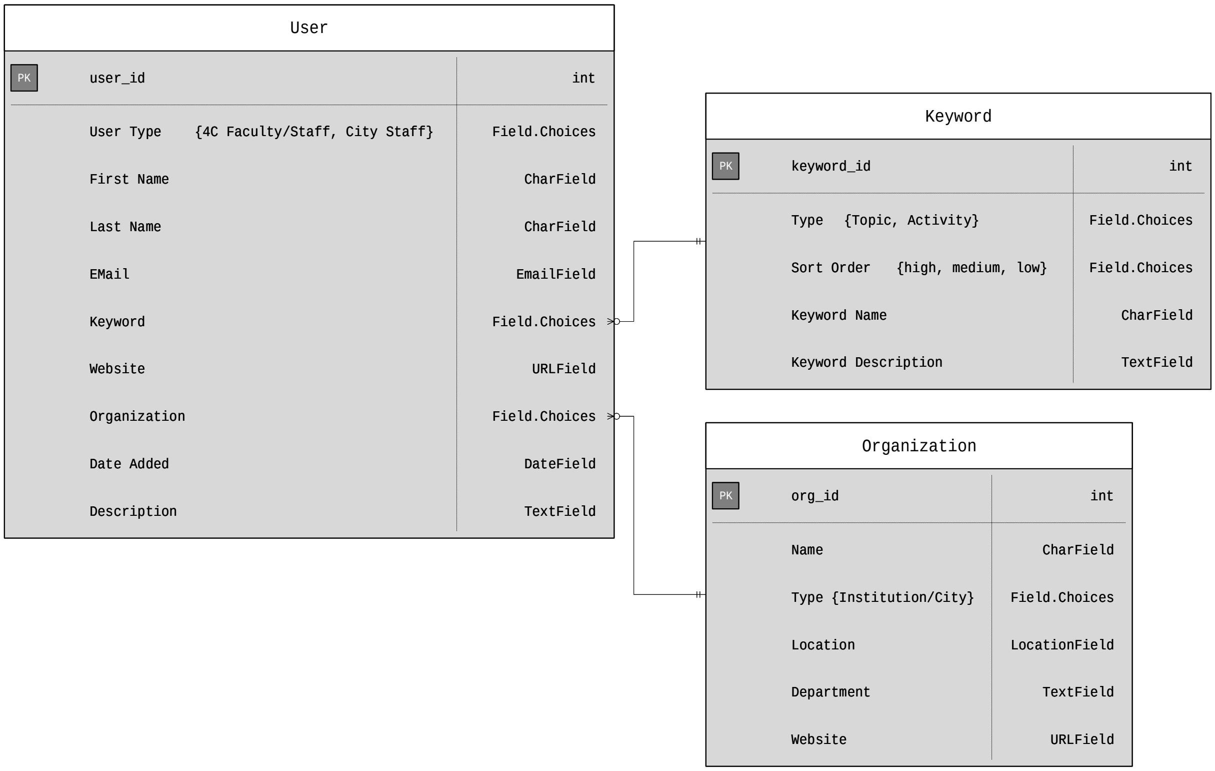
4.2.3 Nginx Configuration

- Serving python through nginx using `WSGI` requires `uWSGI` to be installed in the virtual environment of the production web application environment.
- For more information on Django and Nginx setup, please reference: `https://uwsgi-docs.readthedocs.io/en/latest/tutorials/Django_and_nginx.html`

A Python requirements.txt

```
Django=2.2
```

B Database Design

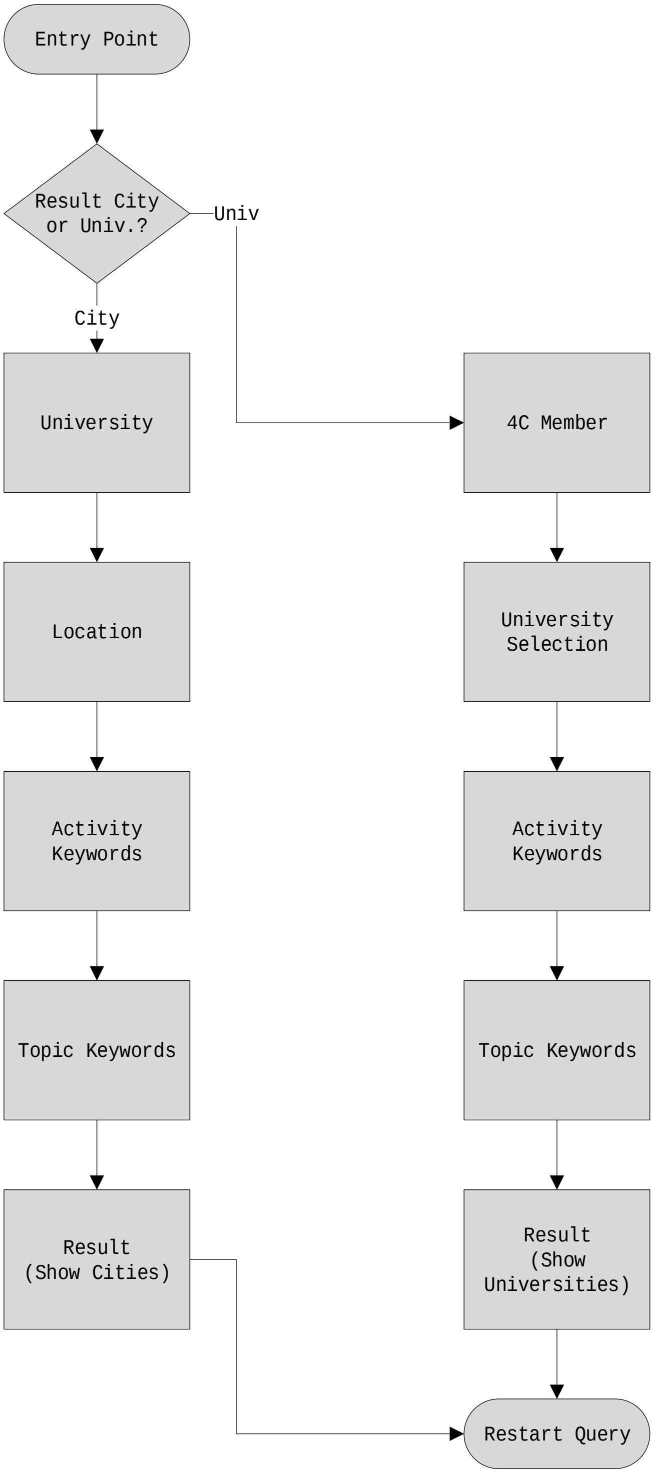


User		
PK	user_id	int
	User Type {4C Faculty/Staff, City Staff}	Field.Choices
	First Name	CharField
	Last Name	CharField
	EMail	EmailField
	Keyword	Field.Choices
	Website	URLField
	Organization	Field.Choices
	Date Added	DateField
	Description	TextField

Keyword		
PK	keyword_id	int
	Type {Topic, Activity}	Field.Choices
	Sort Order {high, medium, low}	Field.Choices
	Keyword Name	CharField
	Keyword Description	TextField

Organization		
PK	org_id	int
	Name	CharField
	Type {Institution/City}	Field.Choices
	Location	LocationField
	Department	TextField
	Website	URLField

C Wizard Data Flow



D Data display

Data will be displayed in a tabular format for both organizations and people. For the organizations view, the name of the organization POC will be displayed along with the organization name, and department. A button will be displayed to show extended information about the organization.

For the person view, the person's name will be displayed along with their department or university. The email address of the person will be displayed. A button will appear to view extended information about the person including, but not limited to their website, bio, and tags.

E API Endpoints

The Django web framework will expose an API so that other services can query all of the keywords (all the keywords, or by keyword type). The API will also have a view for showing all the location regions, and possible departments. It will also allow for viewing of all organizations, and similar to keywords, the user will be able to pick from the types of organizations.
For example, the endpoint my be:

- `<<URL>>/api/getLocations/<<locationType>>`
- `<<URL>>/api/getKeywords/<<keywordType>>`