Note Pond
Software Design Specification

Table of Contents

1. SDS Revision History

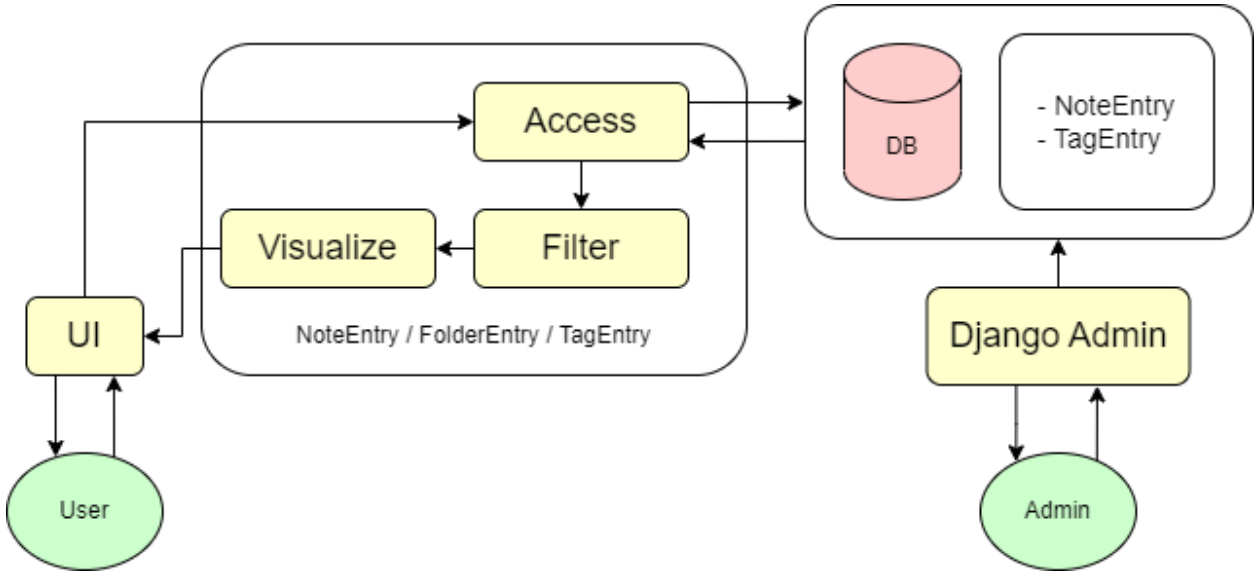
Date	Author	Description
5/16/23	CV	Deleted old content
5/16/23	DH	arch diagram, section 3, started sections in 4

2. System Overview

Note Pond is a digital platform intended for use by students at the University of Oregon. Its main function is to serve as a repository for lecture and study notes. Users of the platform are able to upload their own notes, categorize them by subject or course, and assign relevant tags for easy searchability. This user-generated content can then be accessed by other students who are looking for additional study resources.

The platform is equipped with a search function that allows students to find notes using various parameters, including course names, tags, or specific keywords. After finding the notes they need, users have the option to view them directly on the website or download them for offline

use. Through this, Note Pond aims to facilitate knowledge sharing and collaborative learning among students.



3. Software Architecture

3.1 Architecture Overview
Note Pond has an architecture with 3 modules. The visualize module interfaces with our frontend. The access module interfaces with our database. The filter module exists solely within the backend layer.

3.2 Module Interaction
The 'Access' module interacts with the database to upload and download data, 'Filter' module sorts the data based on user search queries, and 'Visualize' module presents the data to the user in an understandable and accessible format.

3.3 Rationale
This architecture was chosen for its modularity, scalability, and separation of concerns. Each module is independent and can be developed, updated, or replaced without impacting the others.

4. Software Modules

4.1. Access

The module's role and primary function.
Handle uploading and downloading of notes, folders, and accounts to/from the database.

Interface Specification

Dynamic Model
[TODO DIAGRAM] User triggers an action (upload/download), the Access module communicates with the database, performs the action, and returns a response.

Design rationale

Alternative designs

4.2. Filter

The module's role and primary function.
Handle the sorting of notes for access by the visualize module.

Interface Specification

Dynamic Model
[TODO DIAGRAM] User inputs search criteria, the Filter module processes the query, and returns relevant results.

Design rationale

Alternative designs

4.3. Visualize

The module's role and primary function.
Handles the visualization of notes and folders through various views.

Interface Specification

Dynamic Model
[TODO DIAGRAM] User navigates to a specific view, the Visualize module fetches the necessary data and renders the view.

Design rationale

Alternative designs