ETD Collection Management

CS5604 Team 1
Raghav Sethi, Ashrith Reddy Thukkaraju, Hirva Bhagat, Tanya Jain, Wen-yu Lee
S.M.E - Prof. Bill Ingram, Prof. Sung Hee Park

VIRGINIA TECH™

Outline

1. Introduction
    a. Background
    b. Goals
    c. Requirements
2. Approach
    a. Tools and Technologies
    b. Workflow
    c. Architecture
        i. Front-end Page (UI)
        ii. File System
        iii. Database
        iv. APIs
        v. Knowledge Graph
3. Conclusion
    a. Key Learnings
    b. Future Work

VIRGINIA TECH™

Introduction

Section 1

- Background
- Goals
- Requirements

Background

- ETDs are underutilized resources
- Collective goal of CS5604:
  - Develop an **ETD Search Engine**
- Team 1's task:
  - **ETD Management**
- Supporting persona:
  - Curator
  - Experimenter

Goals

- To Provide the necessary services for the Persona **“Curator”**
  - Add single/stream of etds
  - Consistency check of file and database system
  - Statistical reports for collection
- To provide APIs to enable the other teams with the access to the database and filesystem for experimenting with the content and storing the results

Requirements

1. As a curator, I want to add new ETD in the file system and database.
2. As a curator, I want to update data for the existing ETD if I found the data has anything wrong.
3. As curator, I want to check different statistical reports of the current collections.
4. As a developer from team 3, I want to organize the output of the object detection to database and file system.
5. As a developer from team 4, I want to get necessary and well organized data for me to make chapter segmentation, classification, and summarization. Then store them in the database and file system.
6. As a developer from team 2, I want to index all data in the database to set up elasticsearch.

VIRGINIA TECH™

Approach

Section 2

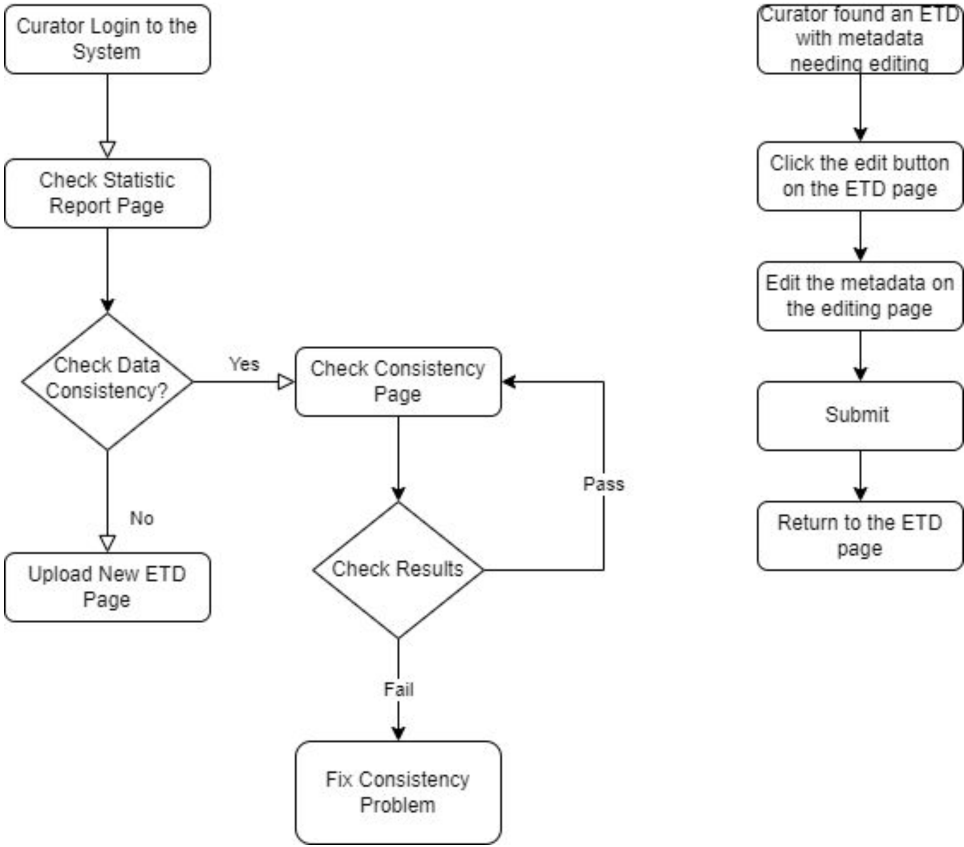
- Tools and Technologies
- Workflow
- Architecture
  - File System
  - Database
  - Front-end Page (UI)
  - APIs
  - Knowledge Graph

Tools and Technologies

- Back-end Framework: Flask
- Database: PostgreSQL
- RDF: Virtuoso
- Database schema design: Vertabelo
- API design and testing: Postman
- VT CS Cloud (cloud.cs.vt.edu)

Workflow

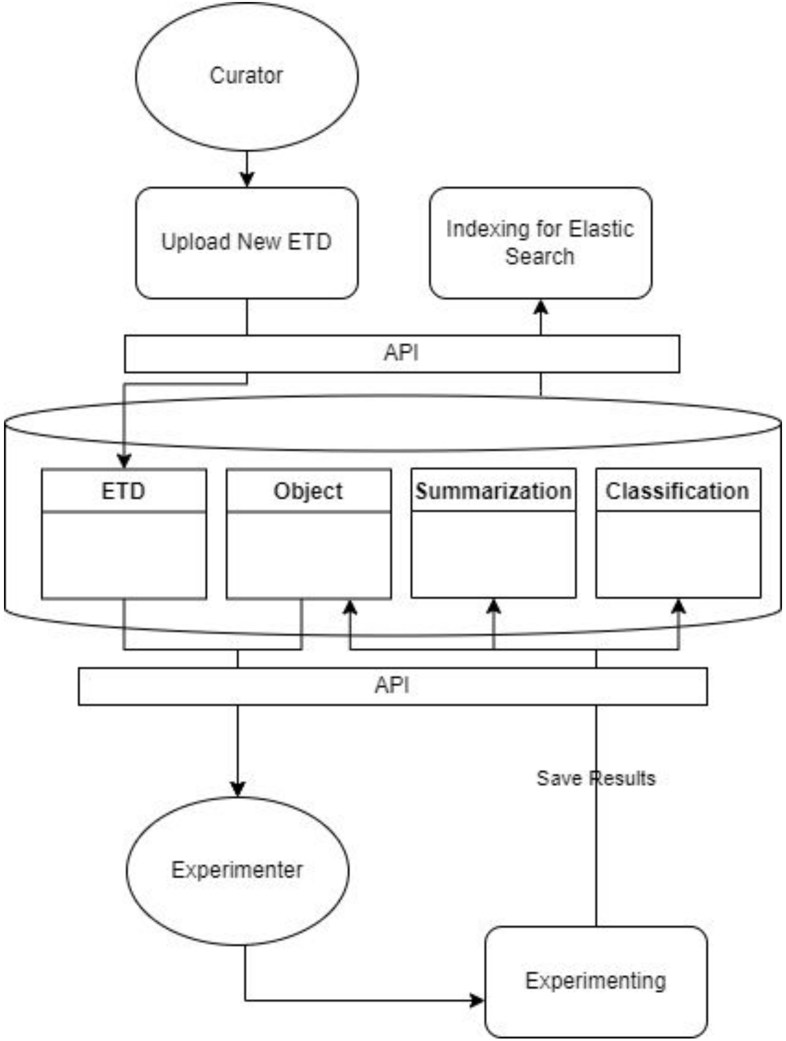
- To provide the necessary services for the Persona “Curator”
  - Add single/stream of ETDs
  - Consistency check of file system and database
  - Statistical reports for collection
  - Edit ETD metadata



VIRGINIA TECH

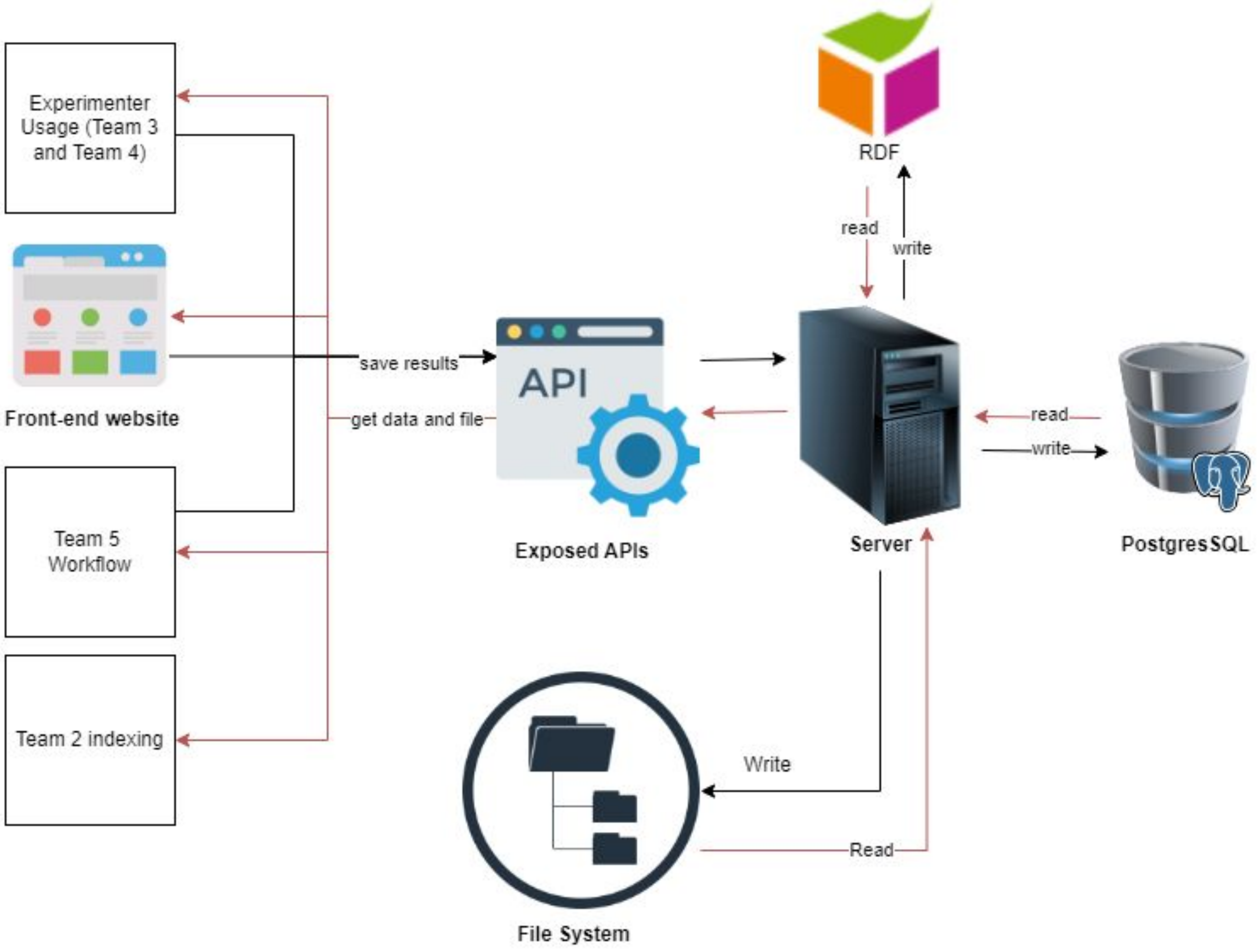
Workflow

- Provide data access for other teams and different services



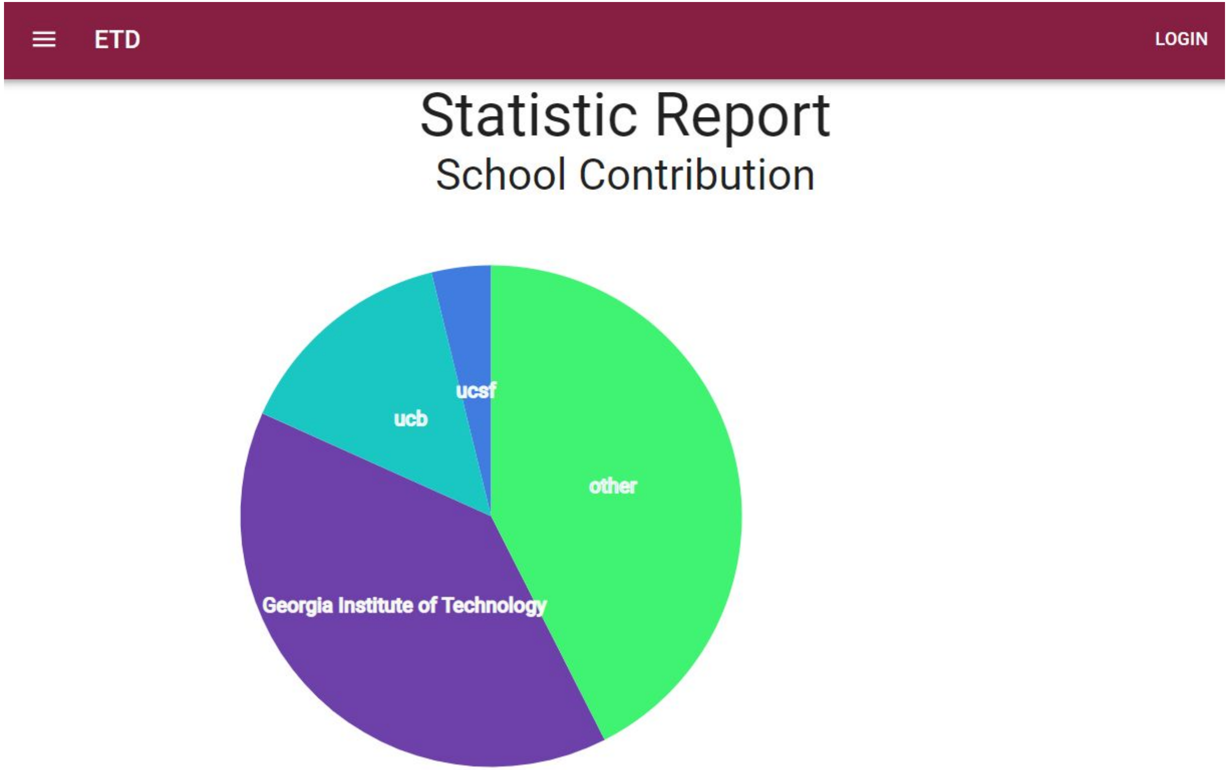
VIRGINIA TECH

Architecture



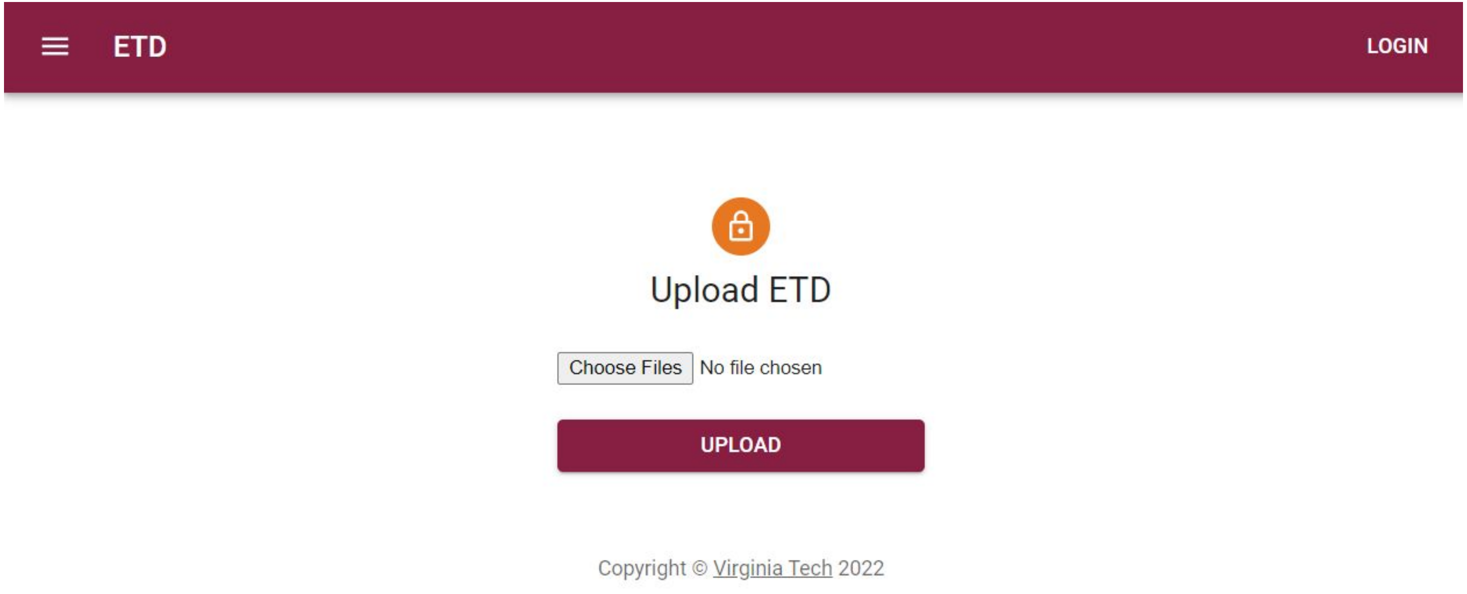
VIRGINIA TECH

Front-End Page - Statistics



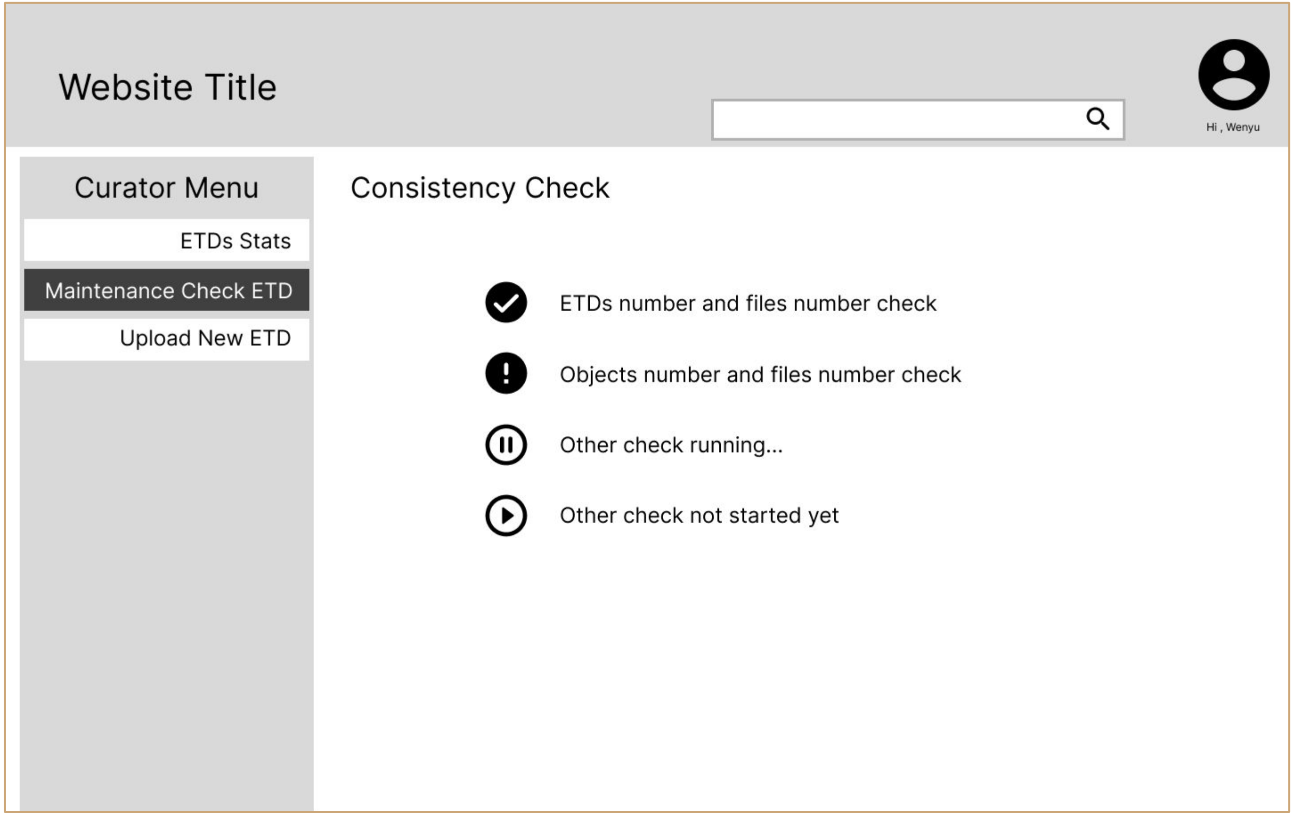
*Based on 50k records of ETDs in the database.

Front-end Page - Upload new ETD



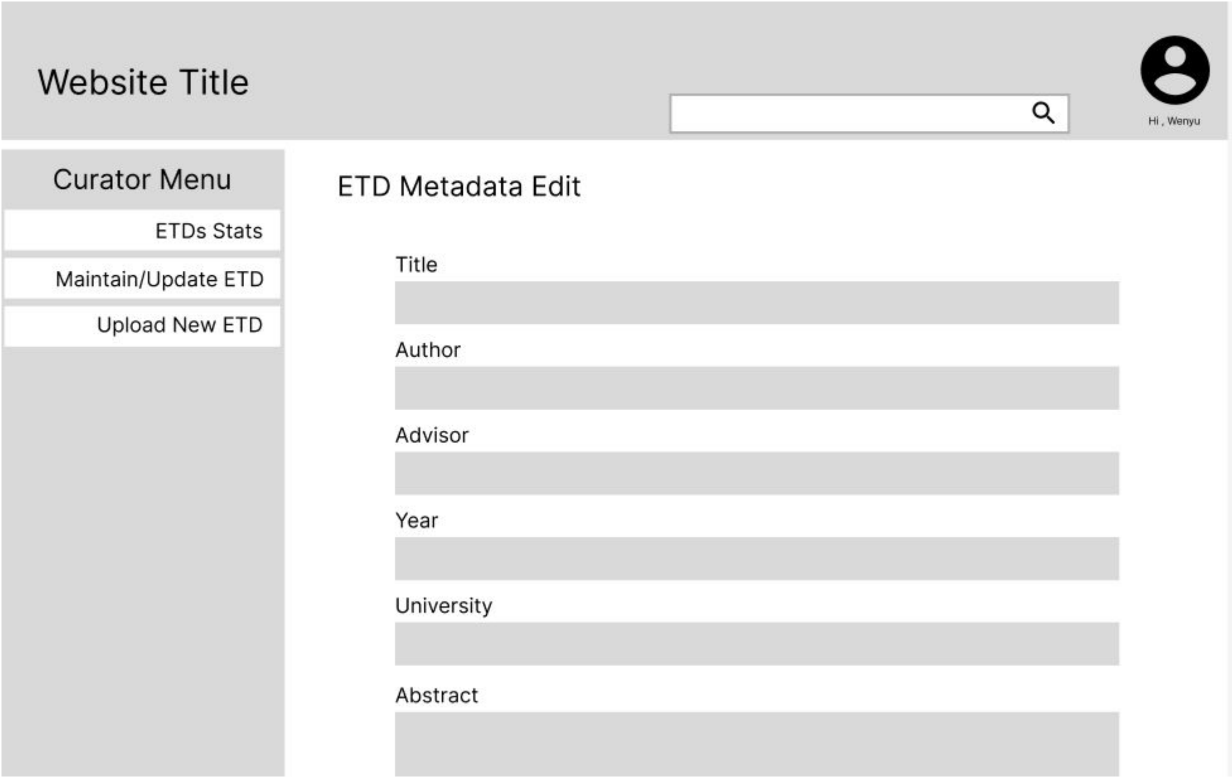
VIRGINIA TECH™

Wireframe - Consistency check



VIRGINIA TECH

Front-end Page - Edit ETD Metadata



Database

PostgreSQL

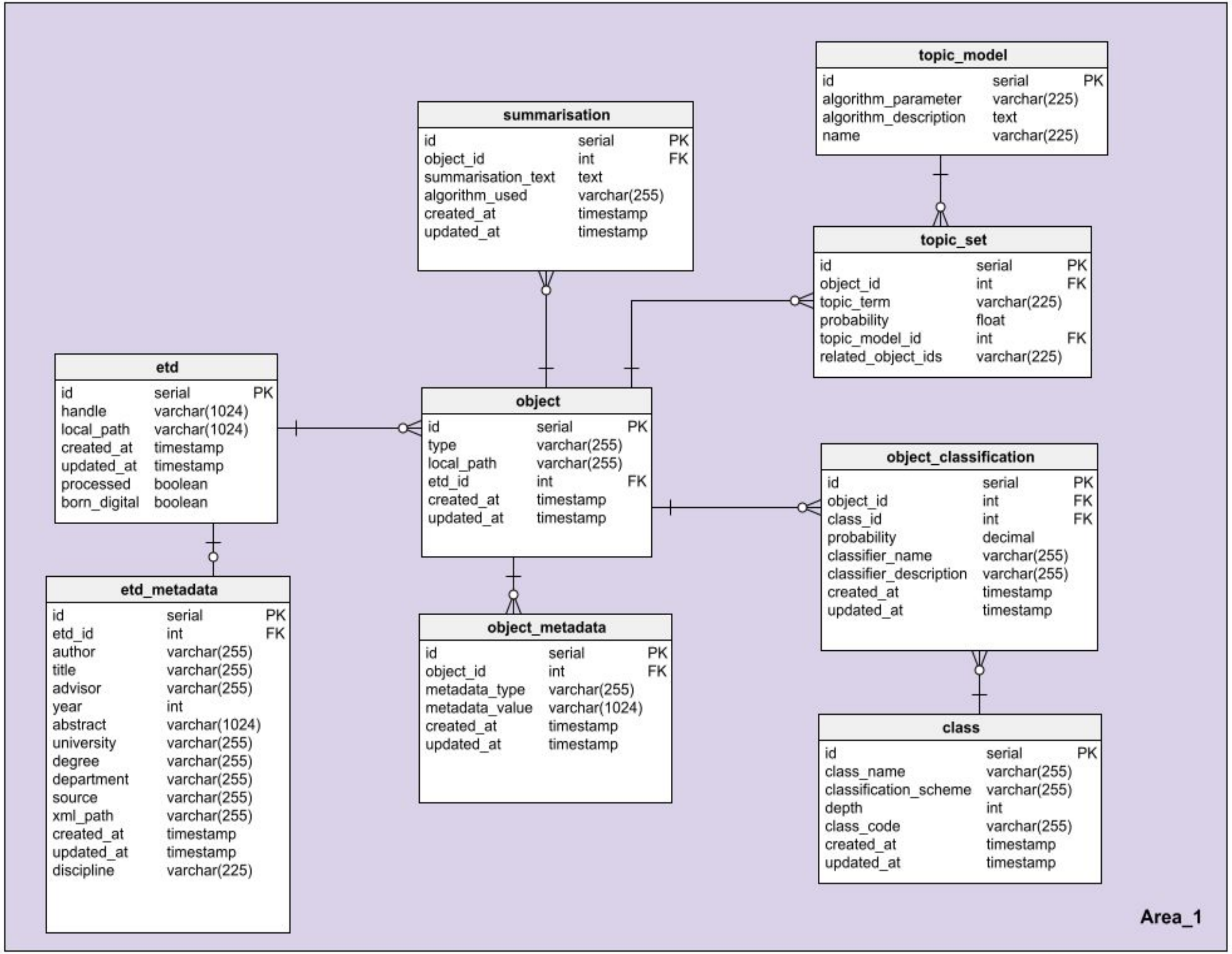
Allows storing large text fields in the DB.

Required for storing **chapter summaries**

DB Schema

Tables

- etd
- etd_metadata
- object
- summarisation
- object_metadata
- topic_model
- topic_set
- object_classification
- class

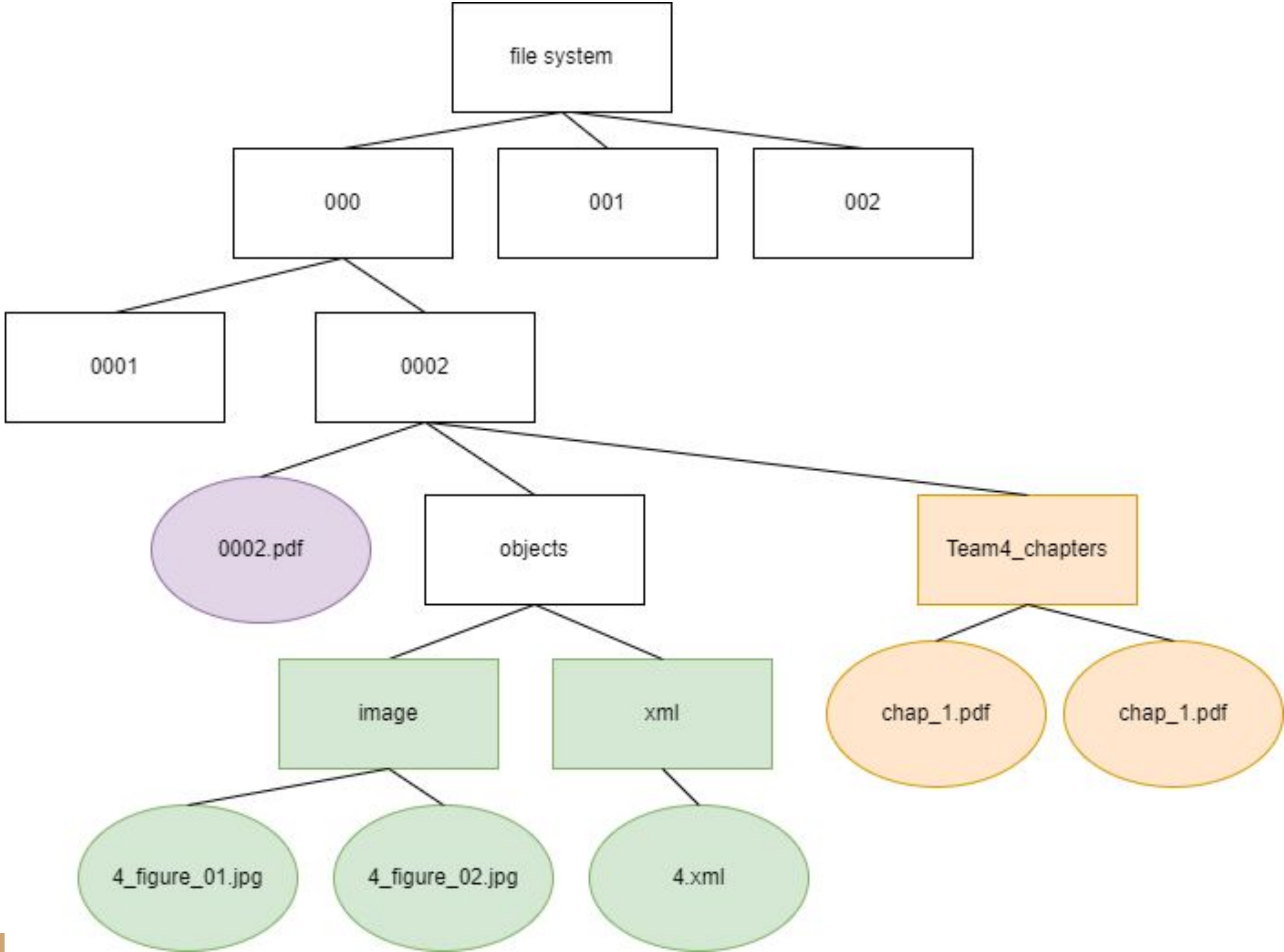


VIRGINIA TECH

Vertabelo

File System

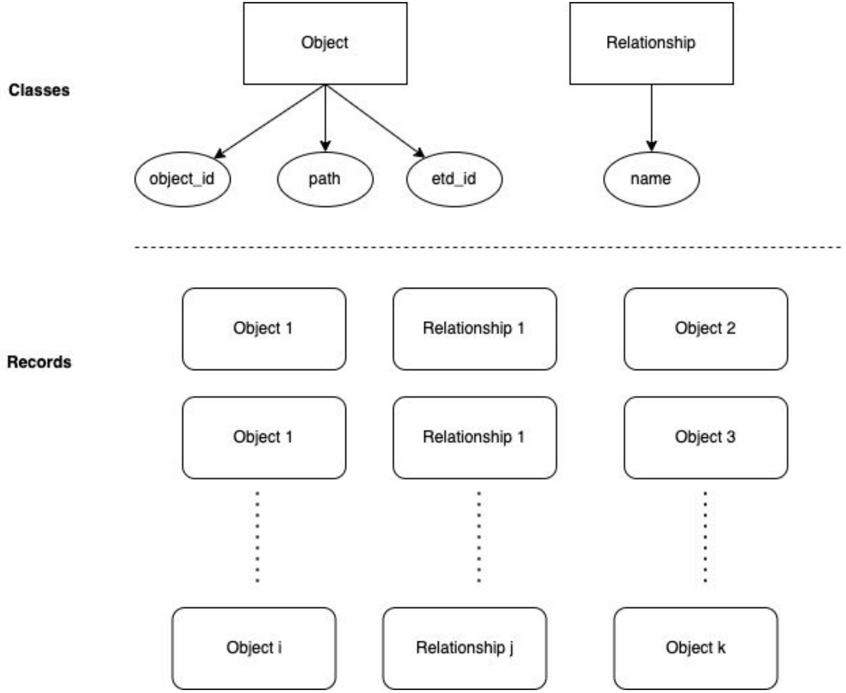
- Hierarchy



API - (Postman Collection link)

- Add new ETD
- Save detected object
- Save detected objects
- Save ETD page images
- Save Topic modeling results
- Save summarisation
- Save classification
- Get all ETDs and objects
- Get ETD by ETD id
- Get objects by ETD id
- Get ETD page image
- Get ETD pdf by ETD id
- Get object file by object id
- Get Statistical report
- Handle and etd-id mapping
- Get knowledge graph by ETD id
- Mark ETD processed

Knowledge Graph




```
// GET /v1/graph/{etd_id}
{
  "relations": [
    {
      "subject": {
        "id": 2,
        "path": "00/001/chapters/chapter1.pdf"
      },
      "relationship": "has_a_figure",
      "object": {
        "id": 3,
        "path": "00/001/figures/figure1.png"
      }
    },
    {
      "subject": {
        "id": 2
```

Conclusion

Section 3

- Key Learnings
- Future Work

Key Learnings

- Understand how different modules work together for ETD Management System
- Dealing with changing user requirements
- Handling data at scale

Future Work

- Knowledge Graph
    - Support Question Answer module
- UI Module
    - Provide more features to Curator, e.g.: Consistency Check
    - Present detailed Statistical Results
- Explore messaging queues to share data
- Enhance backend system
    - Set up dashboard and alerts for metrics, e.g.: API error growth, CPU usage, disk usage
    - Error Handling
    - Set up Kibana to monitor logs

Questions?

VIRGINIA TECH™