NOMCAARRD INTERACTIVE DATA MAPPING DISTRIBUTION AND REPOSITORY

KHENT MARK J. DAHAY

MEREAL KATE C. SILVESTRE

JIU AXL R. TABILLA

UNDERGRADUATE CAPSTONE PROJECT PROPOSAL SUBMITTED TO THE FACULTY OF THE DEPARTMENT OF INFORMATION TECHNOLOGY COLLEGE OF INFORMATION SCIENCES AND COMPUTING, CENTRAL MINDANAO UNIVERSITY, IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY

MAY 2024

Republic of the Philippines
CENTRAL MINDANAO UNIVERSITY
Musuan, Maramag, Bukidnon

College of Information Sciences and Computing
Information Technology

APPROVAL SHEET

The undergraduate capstone project proposal attached hereto entitled, "NOMCAARRD INTERACTIVE DATA MAPPING DISTRIBUTION AND REPOSITORY", prepared and submitted by KHENT MARK J. DAHAY, MEREAL KATE C. SILVESTRE, AND JIU AXL R. TABILLA in partial fulfillment of the requirements for the degree Bachelor of Science in Information Technology, is hereby endorsed.

NEIL JOHN J. JOMAYA Date:
Chair, Thesis Advisory Committee

KENT LEVI A. BONIFACIO Date:
Member, Thesis Advisory Committee

DONAH RAE P. VERULA Date:
Member, Thesis Advisory Committee

Recommending Approval:

EYERYL JUN J. TANO Date:
Department Chair

NATHALIE JOY G. CASILDO Date:
Research Coordinator

Approved:

JOHN D. TAJONES Date:
College Dean

Noted:

JUPITER V. CASAS, Ph.D. Date:
Director for Research

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I
INTRODUCTION

1. Background of the Study

Government agencies depend on accurate and efficient data gathering to assess the development of areas within their country. This data is crucial for planning and implementing programs to enhance development or address issues in these areas. Supporting studies and initiatives highlight the importance of data management and governance in government agencies, particularly in the context of developing organizations. The Urban Institute's research emphasizes the critical role of government decisions and issues in collecting and using data, underscoring the challenges and opportunities inherent in data management (Metzenbaum, Katz, & Nightangle, 2022). This aligns with the broader context of the need for centralized and standardized data collection efforts to enhance the development of areas within these countries.

A report on government data management for the digital age provides insights into the benefits of interoperable and connected government datasets, as well as the importance of offering relevant infrastructure components centrally (Domeyer, Hieronimus, Julia Klier, & Weber, 2021). This report highlights the potential of digital technologies to improve government data management. Moreover, it includes the use of data tracker and consent management tools to enhance transparency and control over how government data is used. These insights could significantly improve the efficiency and effectiveness of data collection and management efforts.

The Philippines, like many developing nations, faces significant challenges in centralizing and standardizing data collection and storing of datasets spanning various disciplines. Based on the analysis of Pasquetto, Borgman, and Wofford (2019), datasets in the field of research were being used for “ground-truthing” and calibration, however issues in finding datasets were one of the concerns. Public repositories were one of the solutions that could cure the lack of data. However, according to Sicilia, Barriocanal, and Alonso (2017), there were already existing repositories for research data; however, challenges in preservation and availability of data exists which includes the minimal control of data uploaded by the users.

Given these circumstances, data mapping emerges as one of the helpful components. Data mapping is the process of linking sets of data from one source to another. It draws and visualizes correlations between their component fields and integrates them into a centralized schema or database for compliance, deduplication, and analysis (BasuMallick, 2022). This method is vital for ensuring that data is accurately transferred and formatted for its new destination, which is essential for a digitalized and centralized repository of data and information. Tools like Geoportal can help overcome these challenges by aiding in accessing geospatial data, thereby promoting strategic planning and decision-making.

According to Utilities One (n.d), by adhering to established data standards and quality control processes, Geoportals ensure consistency and uniformity in data, which is crucial for accurate and reliable assessments of development areas. This standardization facilitates easy data sharing and collaboration among different stakeholders, addressing one of the challenges of managing data from different sectors and regions.

NOMCAARRD is willing to support the development of the NIDMDR. The Northern Mindanao Consortium for Agriculture, Aquatic and Natural Resources Research and Development (NOMCAARRD) is a collaborative organization consisting of 20 government and non-government entities in region 10. These include educational institutions like Central Mindanao University (CMU), Xavier University (XU), and others, as well as government agencies like the Department of Agriculture Regional Field Office-10 (DA RFO-10) and the National Economic Development Authority-10 (NEDA-10). NOMCAARRD's main goal is to coordinate research, development, and extension efforts related to agriculture, fisheries, and natural resources in the region. Its office is situated at CMU.

Despite NOMCAARRD's goals, it faces challenges in effectively coordinating and monitoring its affiliated area. One major issue is the lack of a centralized repository for the data and documents coming from the various parties involved. This fragmentation hinders efficient communication and collaboration among members, potentially impacting the overall effectiveness of NOMCAARRD's efforts. They are receptive to the idea of implementing a centralized repository to prevent data loss and the scattering of information caused by localized storage.

In addition to these practical benefits, the researchers also aspire to gain more public recognition for their valuable findings and discoveries. By showcasing their work through this platform, they aim to increase visibility within the scientific community and the general public. Moreover, they are concerned about the risk of plagiarism in the digital age and desire to protect the originality of their contributions. This platform can offer safeguards against unauthorized use and maintain a traceable record of their work. Overall, the creation of this platform not only addresses the researchers' immediate challenges but also contributes to the broader scientific community's knowledge and public awareness of biodiversity research.

In response to the challenges that were previously stated about dataset scarcity and unavailability, several applications and systems have been developed to address these issues. Notable examples include IncluSet developed by by Kacorri et al. (2020), which serves as a repository for accessibility datasets, EMPIAR by Iudin et al. (2022), the electron microscopy public image archive, National Hydrography Datasets represents the water drainage network of the United States (National Hydrography, n.d.), and Open Data Philippines (ODPH) open data from different government agencies. These systems aim to provide researchers with access to a broader range of datasets and foster innovation in their respective domains.

Furthermore, various platforms are dedicated to managing geospatial data, such as Geoportal Philippines, Global Forest Watch, and the National Wetlands Inventory. Geoportal Philippines facilitates the discovery and access of geospatial information and services (Geoportal Philippines, n.d.). Global Forest Watch monitors the world's forests through satellite imagery (Global Forest Watch, n.d.), while the National Wetlands Inventory focuses on wetland classification and mapping (U.S. Fish and Wildlife Service, n.d.). These systems play crucial roles in providing repositories and monitoring mechanisms for geospatial datasets.

It is essential to acknowledge that existing systems have their limitations. For instance, IncluSet does not offer direct data storage on servers, as noted by Kacorri et al. (2020). While Open Data Philippines cannot be managed directly by the owner of the data. Additionally, Iudin et al. (2022) emphasize the importance of distributors and crowdsourcing to further expand the collection of datasets, recognizing that these systems are not exhaustive solutions on their own.

Notwithstanding the concerted efforts and proposed resolutions by various researchers and organizations, persistent issues persist regarding datasets. Challenges encompassing dataset quality, the efficacy of public repositories, and data scarcity among diverse research cohorts prevail. The collective findings from studies and the needs assessment underscore a necessity for a collaborative platform for diverse and requisite datasets. Furthermore, discussions emphasize the imperative need for a public repository dedicated to storing available datasets rather than confining them solely to localized storage. Collectively, these findings underscore the crucial need for continued efforts to address challenges related to data accessibility, diversity, and sustainability in various domains.

2. Statement of the Problem

This capstone project intends to digitize and centralize the storage of data gathered and researched by NOMCAARRD. The project will focus on mapping and visualizing these datasets for both public and government-related purposes. Specifically, it addresses the following:

1. How will the NOMCAARRD community store its gathered data and information regarding its projects and commodities?
2. How will users achieve data accessibility and transparency while securing the data from unauthorized access and usage?
3. How will projects and commodities be accurately tagged to its corresponding source area?
4. What methods of visualization and transparency will the organization employ for monitoring the projects and development within the area of its jurisdiction?

Moreover, the project will include an evaluation phase to assess the system's performance in meeting these objectives, ensuring it aligns with the needs of the research community.

3. Objectives of the Study

The main objective of this study is to develop a platform for a digitalized and centralized repository for datasets and information. This specifically aimed to:

1. To store data and information through a repository managed by a user or organization.

2. To provide repository creators and managers the ability to configure user access control on their repositories.
3. To integrate tagging of datasets with their source areas.
4. To implement map visualization and historical analysis for the organization to track data pertaining to specific areas.

4. Scope and Limitations of the Study

The project, NIDMDR, seeks to establish a comprehensive repository for datasets, accommodating various data types such as images, CSV files, and text documents. This repository will provide researchers and users with a centralized location to access and manage diverse datasets efficiently. Furthermore, the system will include a geospatial visualization feature, allowing users to visually explore and analyze the geographical distribution of the data. While this visualization will initially be available for region 10, it sets the groundwork for potential future expansions to include more regions.

In terms of security and data validity, NIDMDR implements user authentication through third-party services such as Google and GitHub. This approach ensures that only authorized individuals can access and contribute to the repository, maintaining the integrity and reliability of the datasets. The primary interface for interacting with the system is through a web application, which is designed to be user-friendly and accessible. While the project focuses on web-based interactions, it is designed to be responsive, ensuring compatibility with tablets and smartphones through web browsers, providing flexibility and accessibility to a wider range of users.

5. Significance of the Study

The development of the NIDMDR project will be significant to NOMCAARRD, Central Mindanao University (CMU), and external stakeholders involved in agriculture and related fields. By digitizing and centralizing the storage of agricultural-related datasets, NIDMDR will offer the following key benefits:

- NOMCAARRD:

  NOMCAARRD will benefit significantly from NIDMDR's capabilities. The platform will enhance NOMCAARRD's access to diverse agricultural data crucial for its research and development initiatives. By providing a centralized

repository, NIDMDR will streamline data management processes and enable NOMCAARRD to make informed decisions regarding agricultural practices, resource allocation, and policy development.

- Central Mindanao University (CMU):

  CMU, as the university where the project is being developed, will also benefit from NIDMDR. The platform will provide CMU researchers and students with access to a wealth of agricultural datasets, facilitating research and learning in the field of agriculture. Additionally, CMU can use NIDMDR as a platform for collaborative research and data sharing with other organizations and researchers.

- External Stakeholders:

  External stakeholders involved in research and development of Agriculture, Aquatic and Natural Resources will have substantial impact. The platform's centralized repository and visualization features will enable external stakeholders to access and analyze data more effectively, leading to improved decision-making and innovation in different sectors.

Overall, the NIDMDR project has the potential to revolutionize the way agricultural-related datasets are managed and accessed, benefiting NOMCAARRD, CMU, and external stakeholders alike. By providing a centralized repository and visualization platform, NIDMDR will facilitate collaboration, innovation, and informed decision-making in various fields.

CHAPTER II
REVIEW OF RELATED LITERATURE

The literature review will explore the importance of data repositories in multiple disciplines, the challenges in creating and maintaining such repositories, and the potential of the "IDMR: INTERACTIVE DATA MAPPING DISTRIBUTION AND REPOSITORY" project in addressing these challenges.

1. Review of Related Concept

i. Importance of Data Repository

During the past decades, the research community has expressed their need for a data repository. The need for a centralized repository not only relevant to a single aspect but to multiple research disciplines. In the field of soil science, Samuel-Rosa (2022) stated the need for a centralized and open repository for microscopic images of soil structures and their interpretations. Hou and Pu (2024) also refer to open datasets as a significant factor on drought monitoring.

Not only in soil science, hydrologic studies denote that streamflow data that were publicly available was essential for hydrologic analyses in India (Goteti, 2023). Similarly, the public spatiotemporal water quality dataset of China by Lin et al (2024) supports studies relevant to water quality assessment, modeling, and projection. Comprehensive synthesis of national and state stream and wetland geospatial datasets were needed for informed decisions and managing these resources effectively (Christensen et al, 2022). These highlights the need for comprehensive synthesis and accessibility of such datasets on a national and state level.

Numerous government and national research also affirmed the use of data repositories. Egypt uses online journals and libraries in investigating the strategies and best practices for effective knowledge management (Ahmed, 2024). A return-related information and analysis would be efficient having a repository for data on national return regimes (Elmas, & Mencütek, 2024). In the medical field, there were also various studies stating the importance of repository. Männikkö et al (2024) stated that

a Patient Data Repository shows the source of development and evaluation of medical risk calculators.

Moreover, in the development and application of artificial intelligence (AI) algorithms, repositories for image datasets play a pivotal role. Few fields that use image datasets are: development of AI algorithms (Zhou, Lapedriza, Khosla, Oliva, & Torralba, 2017), object detection (Nguyen et al, 2023), medical image analysis (Ying et al, 2023), radiology-based machine learning tasks (Sabottke & Spieler, 2020), and crop disease image recognition (Yuan, Chen, Ren, Shimei, & Li, 2022). However, despite their importance, several studies and reviews on different systems stated that lack of image datasets and data scarcity as one of their numerous problems.

Despite the recognized need for a centralized repository of research data, various challenges persist. Traditional practices of saving data in disparate locations and formats hinder the establishment of a unified and up-to-date repository. This fragmentation not only complicates access for researchers but also limits the effectiveness of data-driven analyses and decision-making processes. Addressing these challenges requires concerted efforts to standardize data formats, enhance data sharing practices, and establish mechanisms for continuous updates and maintenance of the repository.

ii. Existing Repositories

In the realm of data repositories, numerous pre-existing databases have served as fundamental resources extensively utilized by researchers in their respective field of study. These repositories have played a crucial role in enabling advancements in various domains. However, recognizing the constraints and gaps within these existing systems, there has been a surge in initiatives proposing and developing new repositories. These innovative ventures have specifically aimed to address the limitations of current repository structures.

In the Philippine government, the major curation related to the access of datasets was the Open Data Philippines. ODP creation was caused by the partnership to Open Government Partnership. ODP aims to provide more accessible government data in a single portal and open format. However, challenges facing open data in the Philippines include limited citizen involvement and access. Moreover, a lack of culture

of openness, and the need for comprehensive government-wide policies and awareness campaigns (Pacis, 2017). This underscores the importance of fostering a culture of openness and transparency in government data management.

In Russia, SibMed Clinical Data Repository was made for the facilitation of research and development in healthcare. Including the development and testing of new digital products, and data analytics in diagnosis and treatment (Kulikov et al., 2023). Meanwhile in Korea, they established the Korean Nucleotide Archive (KoNA) to collect and manage raw next-generation sequencing data from national genome projects (Ko et al., 2024). KoNA is a national repository to deposit biological data generated from government-funded research projects in Korea. Unlike SibMed, KoNA accepts deposit or submission of biological data from users using GBox, and Bio-Express.This demonstrates the global trend towards developing specialized repositories to meet specific research needs.

In the agricultural sector of India, they developed a centralized data repository system of Indian Council of Agricultural Research (ICAR) named Krishi (Biswas, 2023). It consists of Technology, Data generated through Experiments/ Surveys/ Observational studies, Geo-spatial data, Publications, Learning Resources etc. Repositories also play a part on tourism, Crete created Crete Destination, a repository provides access to cultural and touristic content (Partarakis et al., 2023). These developments reflect a trend on sharing diverse types of data and resources to support research, decision-making, and public access in respective fields.

Accessing data and datasets of different disciplines, Harvard Dataverse stands as one of the created curations. This repository allows for the sharing, archiving, citation, access, and exploration of research data (Harvard Dataverse, 2024). Each Dataverse collection is a customizable, dataset-focused repository designed for organizing, managing, and highlighting datasets. Depositors can add dataset-level metadata and use free-form keyword fields, but they cannot add their own detailed file-level metadata. This limitation might restrict the ability of researchers to fully describe and categorize their data at the file level.

Moreover, in the landscape of pure image datasets for AI and machine learning, Fei-Fei et al (2009) introduced a study that creates and formulates image datasets.

ImageNet stands out as a prominent repository, boasting over 14 million images categorized into 1000 classes, facilitating robust image classification and recognition tasks. A work of Lin et al (2014), marked as another significant repository which was COCO (Common Object in Context) formulated by large-scale object detection, segmentation, and captioning dataset. This serves as a solid reference point for evaluating other types of labels, such as scene types, attributes, and full sentence descriptions.

By that, the importance of data repositories cannot be overstated, as they serve as fundamental resources that enable advancements in various fields. Given the crucial role of repositories in organizing, managing, and sharing data, it is essential for organizations to create and maintain them. These repositories not only facilitate research and decision-making processes but also promote transparency, accessibility, and collaboration in the academic and scientific communities.

2. Related Systems

Geospatial data plays a crucial role in accessing repositories by providing spatial context to the information stored within them. It enables users to visualize and analyze data based on its geographical location, allowing for better understanding and decision-making. The advantage of developing such a Geo-portal is to eliminate redundancies and duplication of efforts, and enforcing consistency, standards, and shareable protocols (ICAR-NBSS&LUP, n.d.).

Global Biodiversity Information Facility (GBIF) is an international archiving system that integrates mapping for locating a specimen of a specific area. By mapping the locations where species have been recorded, GBIF can provide a clear picture of where different species are found around the world. This is essential for understanding the geographical range of species and identifying areas of high biodiversity (GBIF.org, n.d.). There is also Global Forest Watch (GFW) who uses mapping of forests for monitoring and alert systems designed to empower people everywhere with the information they need to better manage and conserve forest landscapes (UNEP, n.d.). Both GBIF and GFW support publications of datasets, GBIF publishes using request endorsement while GFW shares via GFW’s crowdsourcing tools, blogs, and discussion groups.

The Philippine government also strives to incorporate mapping of information through Geoportal Philippines.The Geoportal Philippines supports the usage of conventional multiscale basemaps as tools for strategic planning, decision making, situational analysis, and other common needs (GeoportalPh, n.d.). Concurrently, Krishi which repository and visualization of Indian Agricultural Research Information integrates Bhoomi Geoportal. This enables visualization of various point, line and polygon thematic layers on soil and land resources (ICAR-NBSS&LUP, n.d.). Both stated geoportals visualize information from the data that submitted by their affiliated agencies and organizations.

The integration of geospatial data is crucial in accessing data repositories. It provides spatial context to the information stored, thereby allowing users to visualize and analyze data based on geographical locations. This ultimately leads to better understanding and enhanced decision-making. Notable examples of this integration include the Global Biodiversity Information Facility (GBIF) and Global Forest Watch (GFW), which utilize mapping to provide clear pictures of global species distribution and forest conservation efforts. The Philippine government's Geoportal and India's Bhoomi Geoportal also exemplify the use of geospatial data in strategic planning and decision-making. These platforms visualize information from data submitted by their affiliated sectors and organizations, further highlighting the significance of geospatial data in providing access to repositories.

The “Interactive Data Mapping Distribution and Repository” capstone project aims to address the lack of a unified platform for sharing and accessing data in NOMCAARRD. This project is designed to be a repository for storing data that they have gathered and new information created associated with the organization. Through the collaborative sharing of data and information by different affiliated organizations, it aims to provide data representation and interactive map visualization. NIDMDR is intended to serve allies of NOMCAARRD by providing a platform for centralizing knowledge exchange and collaboration.

By far, no specified platform is intended to showcase data presentation and visualization of the achievements and discovery of NOMCAARRD. This limits the association to organize and arrange the data from the numerous affiliated organizations. However, through the guidance of the mentioned studies and systems,

it serves as a guide to create, develop, and deploy IDMR. The development of this platform aspires to organize data and information that will lead to better monitoring and collaboration of every partnered area.

CHAPTER III
TECHNICAL BACKGROUND

The Interactive Data Mapping Distribution and Repository aims to digitize and centralize the storage of datasets collected by NOMCAARRD. This platform will enable organization members to establish repositories, categorize datasets into specific geographical areas on a map, and visualize data. Several web development frameworks, libraries, programming languages, and software development tools will be used in the development of the project. These tools will enable the proponents to achieve the objectives stated in the capstone project.

Vue.js will be utilized in the front-end development of the entire project. It is designed for creating user interfaces and single-page applications (SPAs), which are applications delivered to the browser without requiring page reloads during use (Mikowski et al., 2013). Vue.js will serve as the backbone of frontend development in this project because of its advantages over other JS frameworks, including its lightweight size and superior performance in applications requiring constant view rendering (Kumpulainen, 2021).

The proponents will utilize Vue.js in creating and rendering the user interface and handling the user input. The Composition API in Vue.js will be leveraged by encapsulating component logic into reusable functions, enabling efficient logic reuse, and maintaining the state and logic of the components in an organized manner. (Djirdeh, et al., 2018). The framework will receive the inputs of the user and will pass it to Axios. Axios is a promise-based HTTP client that provides an easy-to-use API for making asynchronous HTTP requests to REST API endpoints in browsers. It is often used as an alternative to the native JavaScript Fetch API due to its additional features and capabilities. (Innocent, 2023).

The proponents will leverage Django as a backend to create Representational State Transfer (REST) APIs using the Django REST framework. Django is a server-side web framework based on Python. Web frameworks like Django include libraries that handle common tasks such as database creation, and querying, thus reducing the amount of tedious and repetitive tasks that developers typically undertake (Bennet,

2008). Django will be used to create REST APIs. These APIs will serve as the communication layer between the front end and back end. Axios will be used to facilitate the exchange of data between the front end and the back end. Furthermore, PostgreSQL will serve as the database system of the project. As Vue.js sends HTTP requests using Axios to Django, Django will interact with PostgreSQL to store, retrieve, and manipulate data as required by the application.

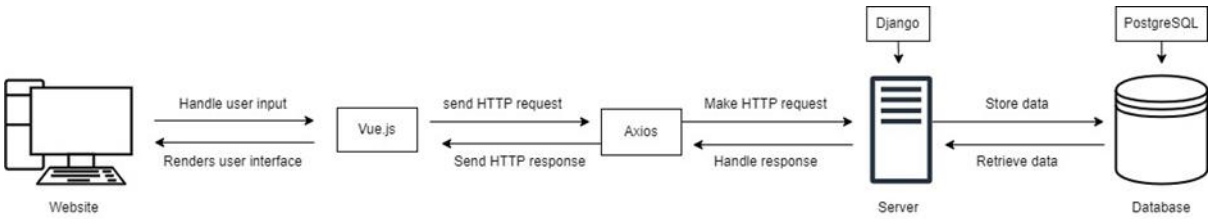

Figure 1. Project Interaction

Figure 1 shows the representation of the process in which the tools stated that will be used in the project interact. It shows how the request is being processed and how the technical tools participate in that process. This process will be seen repetitively in achieving the objectives of the project. Also, these tools have their specific role in realizing the objectives of the system.

NOMCAARRD faces challenges in managing and sharing the datasets it collects, particularly in making this data accessible to sectors associated with or under the organization. To address this problem, the project team plans to create a centralized repository module. This module will facilitate collaboration, data storage, and sharing among organization members. Additionally, the team will implement user access control to vary access levels based on members' roles, ensuring that only authorized individuals can access specific data.

It will utilize Vue.js to create the user interface of the module. This interface will allow users to view, create, modify, and delete repositories. It will also include the interface for requesting access to the repository allowing dataset collaboration, and for the uploading of the datasets. On the other hand, Django will be utilized in this module to handle server-side operations. Django will also handle functions such as compressing files for downloading purposes and decompressing files if the user uploads files in batches. When the logic for these functionalities is created, REST APIs

will be created for Vue.js to send requests to the APIs with the help of Axios. For the storing of data, PostgreSQL will be used by the proponents. Django will interact with PostgreSQL in storing the repositories and the datasets.

The repository owner can establish access controls and configure the repository's visibility. If a repository is designated as public, it will be accessible to guest users who are not part of the organization. This functionality will be implemented by creating a module for public users to access data and information from publicly disclosed datasets by relevant organizations. Guest users, even those not affiliated with any organization, can access public repositories. This process will occur in the backend, where the backend will query the database for repositories made public by certain organizations. The data retrieved will then be sent to the frontend and displayed on a specific page, making it accessible to guest users.

The proponents will also integrate map visualization in the project to monitor and track data about specific locations. The proponents will resort to Maplibre, for displaying the provinces of Region X on the module. The choice of MapLibre over other tools such as Google Maps API, which is proprietary and requires billing, is primarily due to its open-source nature, which allows for greater flexibility and customization.

The boundary of each province of Region X will be highlighted and divided. To highlight the boundaries, the proponent will utilize geoJSONs which is a format for encoding geographic data structures. These geoJSONs will be stored in PostgreSQL. PostgreSQL will serve as the database for NIDMDR due to its extensibility and support for various data types such as JSON, which facilitates efficient storage and retrieval of complex data, thereby enhancing system performance (Riggs, et al., 2019). The choice of PostgreSQL for this project is also due to its ability to handle geospatial data and its capabilities for storing and managing geoJSON formats (Hsu, et al., 2015).

This module will feature two types of map visualizations. The first involves the display of predefined locations, such as the provinces in Region X. The second type allows users to create custom locations on the map. To implement this, the project team will utilize the mapbox-gl-draw, a JavaScript library that enables users to draw custom shapes and polygons across the map. This library will be integrated into the

frontend. Users will have the option to either upload a GeoJSON file for a custom boundary or draw a boundary directly on the map. The coordinates of the drawn boundaries will be sent to the Django backend for storage in the database.

The primary function of the creation of map visualization module is to display repositories located within specific areas, including both built-in locations like the provinces of Region X and custom boundaries created by users. To achieve this, the proponents will implement dataset tagging. When a user imports or creates a custom boundary, a form will appear, prompting the user to tag a repository to that boundary. Upon completion, Vue.js will send a request to the backend using Axios. Django will then validate this request and return an error response if necessary. If the request is successful, the data will be stored in the database, and Django will notify the frontend, indicating that the request was successful and redirecting the user to the page where the repository is now associated with the sector.

The tools, technologies, and frameworks mentioned are essential for realizing and achieving the objectives of the study. These technical resources will significantly aid the proponents in successfully executing the capstone project. Additionally, the integration of these tools will not only streamline the development process but also enhance the project's functionality, security, and user experience. This comprehensive approach ensures that the project not only meets the technical requirements but also provides a robust platform for its intended users.

CHAPTER IV
METHODOLOGY

1. Conceptual Framework

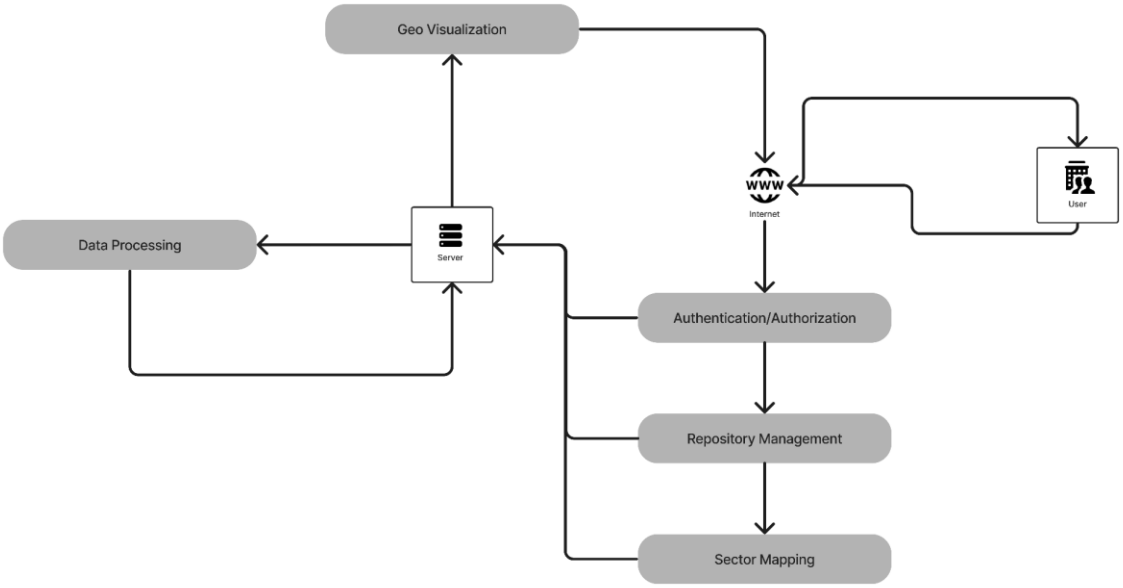

Figure 2. Conceptual Framework

This platform will operate through an architecture comprising four key modules: authentication, repository management, sector mapping, and geo visualization. Each module will play a distinct yet interconnected role in orchestrating functionality of the platform. The conceptual framework for the development of a digitalized and centralized repository for datasets and information will encompass several key components. Firstly, the system itself will serve as a central hub for storing, organizing, and managing datasets and information. This will include implementing measures for security and transparency, making data access and usage clear and understandable to relevant parties.

Secondly, this will involve the implementation of data tagging, which will be essential for associating information with specific areas. This process will categorize and identify datasets based on their source or content, facilitating easy retrieval and

use of information. Additionally, this will include mechanisms that will enable or disable public access, ensuring that datasets and information published are accessible to the general public.

Moreover, it will incorporate the use of geo visualization, which will involve the visualization of data on maps to provide spatial context. This will allow users to see data represented visually, enabling them to better understand patterns, trends, and relationships within the data. It will deal particularly in viewing and accessing data pertaining to specific areas under the organization's jurisdiction. Overall, it will provide a structured approach emphasizing security, data tagging, transparency, and geo visualization as key components.

2. System Development Methodology

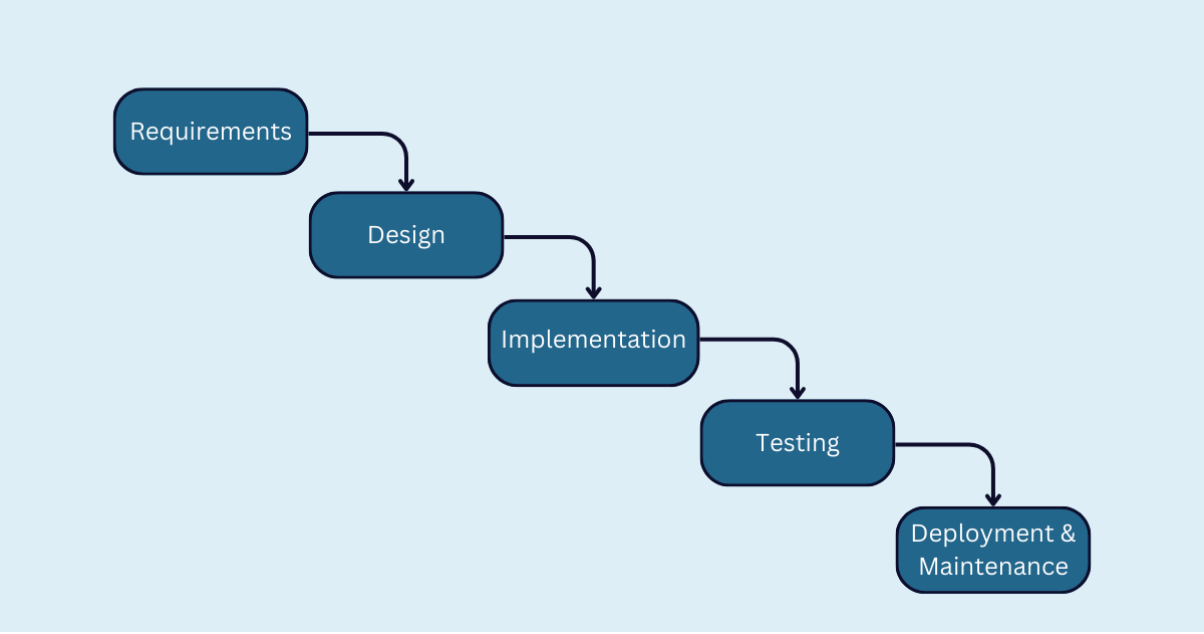

Figure 3. Waterfall Model

Due to several reasons, the proponents choose Waterfall methodology to be used for the development of Interactive Data Mapping Distribution and Repository. Firstly, during the needs assessment and requirement analysis, the client's objectives for the project will be needed to be clear and well-defined. This will unlikely to change during the development process which makes the waterfall model suitable. Upfront planning and sequential approach has been done to ensure the system stays on track (Khan & Mahadik, 2022).

Additionally, the Waterfall model will allow for early identification and mitigation of risks by defining requirements and design upfront. This will reduce the likelihood of costly changes later in the project, as potential issues will be addressed early on. NOMCAARRD's involvement will be strictly required during requirements definition. It will be beneficial that clients have a clear idea of what they want and prefer a more hands-off approach during development.

i. Requirements

Through a series of interviews with stakeholders, the proponents will be able to understand the current process of the system utilized by the clients. The interview will include how types of data are formatted and stored, how such data is utilized for data analysis or decision-making processes, and the steps taken to secure data transfer process from source to storage. The proponents will also include any parties involved during the process and what are their roles and responsibilities in coordinating with NOMCAARRD.

Once the interview process has been completed, the proponents will then assess how to simplify the process and find strengths and weaknesses through each step as detailed by NOMCAARRD. Furthermore, the proponents will also assess how to make data storage, retrieval, and presentation be organized such that the client and its associated parties would be able to utilize the system in a secure and efficient manner.

ii. Design

Building upon the established requirements, the proponents will embark on the design phase to create the system architecture. The design process will involve defining the overall structure of the platform, including the database schema and system modules. Detailed database designs will be crafted to ensure efficient storage and retrieval of datasets. Additionally, the design phase will address the integration of map visualization tools. The resulting system design document will provide a blueprint for the development team to implement the envisioned features and functionalities.

Based on the objectives specified during the requirements, the proponents will then design flowcharts integrating the processes that need to be done to achieve the component's goal. At this phase, the proponents will create a database schema starting from a generalized database up to a normalized database. Moreover, prototyping for the specific module will be made to visualize and validate the design components that the sprint will implement.

The database schemas designed will be implemented in the PostgreSQL database system. The reason behind this preference is its adaptability and its capacity to accommodate diverse data formats like JSON, enabling effective storage and retrieval of intricate data, consequently improving the system's performance (Riggs et al., 2019). Additionally, PostgreSQL was selected for this project because of its competence in managing geospatial data and its adeptness in handling storage and management of geoJSON formats (Hsu et al., 2015).

iii. Implementation

With the design in place, the development team will begin implementing the platform. This will involve writing code to create the database schema, develop the user interface, and integrate the map visualization and dataset tagging features. The implementation phase will also include unit testing, where each component of the system is tested in isolation to ensure it meets its specifications.

iv. Testing

Once the platform has been implemented, it will undergo two phases of testing to ensure that it functions as intended. The first phase will include alpha testing, divided into two phases, (1) integration testing and (2) system testing. Integration testing evaluates whether every component of the system supplements and complements one another, and system testing, where the entire system is tested as a whole. The testing phase will also include user acceptance testing, through System Usability Scale survey, where

stakeholders and selected participants will use the platform to ensure it meets their requirements.

v. Deployment and Maintenance

After the platform has been deployed, the maintenance phase will begin. This will involve monitoring the platform for any issues or bugs and addressing them promptly. The maintenance phase will also include making updates to the platform to add new features or functionality based on user feedback and changing requirements.

3. System Analysis

i. Flowcharts

Figures 5-8 from Appendix A show the process of the registration and login. NIDMDR allows users to connect, register, and login via different third-party services, specifically the GitHub and Google authentication. This concept involves linking a user's identity and associated attributes across various individual identity management systems. When users log in using Google, GitHub, or other platforms, the third-party service acts as an identity provider, confirming the user's identity to the website or application seeking authentication. Furthermore, the validation error might occur due to the unregistered inputted information which will indicate the need for registering as a new user.

Furthermore, login with Google and GitHub features starts with the frontend side where the client sends requests to the Google and GitHub authentication API. Google and GitHub will follow their respective authentication and authorization process and after a successful authentication these services providers will send a secret and authentication key. These will be used by the frontend to send a request to the backend server. These tokens will then be used by the backend server to communicate with the service providers to get user details and authenticate the user on the backend side. After a successful authentication with the backend server an auth token and a refresh token will be generated and will be sent to the frontend client. These tokens will be used in every request to identify the user.

Figure 9 to 11 of Appendix A shows the flowchart for creating, reading, updating, and deleting data from backend and frontend. General overview from figure 9 shows decision point for reviewing new data. If there is no new data, the backend will read existing data and will transmit to the frontend and be viewed in the interface. Otherwise, new data might be caused by creating, updating or deleting data which happens in the backend and will be presented on the frontend.

Figure 10 and 11 shows the add, update, and delete process of the system. It started from users browsing to the platform, then input new data, update existing data or sending requests of deletion. Those data will be validated and if those data are valid, it will be sent to the backend then processed and stored. Lastly, the backend will send a response to the frontend, enabling users to know if the action was done successfully.

Moreover, figure 12 shows the process of sector tagging. Users can select a specific sector and associate it with an entity. The entity will be validated and will display a message to the frontend. The last part of the process is showing the updated data to the frontend, showcasing new sector-entity associations to the map.

ii. Data Flow Diagram

The figure 13 from Appendix B depicts the context data flow diagram, illustrating how data flows within NIDMDR. The diagram reveals that the system receives and sends out data from two entities which are organization owner and organization member. Organization owner has the capability to input the organization data, access control data, and receive the organization information. While organization members can add repository data in the system and can access repository information and access control level information. The incoming data represents the initial input to the system, while the outgoing data signifies the final output from the system.

Appendix B figure 14 shows the level 1 of the data flow diagram of a user. Users have access to five processes which are authentication & authorization, repository management, sector management, access control management, and file management. Users initiate the authentication and authorization process by utilizing their credentials to acquire an access token. For repository management, the system requires repository data, which is then stored in the database. In the case of sector

management, users can input GeoJSON data into the map interface, which is subsequently stored in the database and retrieved for display on the interface.

Access control management allows users to define and associate access controls with their respective repositories. Lastly, the file management process handles file data, storing it in the database and providing file information back to the user. All data used in these processes is stored in the database, and its presentation in the interface is determined by the access controls set on the data.

iii. Use Case Diagrams

Figure 15 from Appendix C presents the use cases under the Organization Owner. The actor involved in this use case is the Organization Owner. This figure shows the interaction of the actor and its role in the system. Organization Owners have the ability to add, and remove members to the organization. It also allows managing of access control to users in an organization.

Figure 16 from Appendix C showcases the general use cases that both the Organization Owner and Organization member can perform within the system. These actors are involved in use cases such as authentication, profile management, and repository management. It implies that organization members cannot add, remove and manage access control to the other members of the organization.

Figure 16 also presents the functionalities available to guest users within the system. In this system, a guest user refers to someone who isn't logged in or authenticated. Their capabilities are restricted compared to those of an organization member. A guest user can only access, filter, search repositories and download files in the repository if the repository owner has set the repository visibility to public. Similarly, with map data, a guest user can only view specific sectors if the creator of that boundary has made it public.

4. System Requirements Specifications

i. Functional Requirements

In appendix D, the proponents have identified 27 crucial functional requirements tailored to meet the needs of its users. These requirements are vital components that must be integrated to effectively realize the objectives of the study.

ii. Non-Functional Requirements

a. Security

Authentication

The system will implement secure login mechanisms using Google, Gmail and GitHub accounts as well as protecting user data during the registration and profile management process.

Password Hashing

All user accounts will have their password hashed using PBKDF2 algorithm with SHA256 hash.

b. Performance

User Traffic

The system must be able to handle at least 50 simultaneous users at once. The system must be able to scale its resources to accommodate peak user traffic.

c. Maintainability

Documentation

The system should have comprehensive and up-to-date documentation that includes information about the system implementation, design, and structure to ensure an easy understanding if the system needs modifications.

iii. Other System Requirements

a. Minimum Hardware Requirements

PC

For other system requirements, the minimum hardware specifications include a PC with at least a dual-core CPU and 4GB of RAM. For the processor, a dual-core CPU ensures that the system can handle basic multitasking and processing demands effectively. The 4GB RAM requirement is to ensure that the system has enough memory to run the operating system and applications smoothly without experiencing significant lag or slowdowns.

Network Requirements

In terms of network requirements, a bandwidth of at least 1 Mbps and latency under 100ms are necessary. A bandwidth of at least 1 Mbps is necessary to ensure that the system can communicate effectively with servers and other devices on the network. A latency of under 100ms is important for ensuring that there is minimal delay or lag in communication, which is especially important for real-time applications or services.

b. Minimum Software Requirements

Browser

For the minimum software requirements, the system should have a compatible web browser such as Chrome version 92.0.4515.159 64-bit, Firefox version 91.0.2 64-bit, or Microsoft Edge version 92.0.902.84 64-bit. These specific software requirements are recommended to ensure compatibility and optimal performance with the system. Regarding the web browser versions, it's important to specify the minimum versions to ensure that the system's web-based components and features are supported and function correctly.

Operating System

As for the recommended operating system, Windows 10 is suggested because it is a widely used and stable platform that provides good performance and compatibility with a variety of software applications and hardware devices. Windows 10 offers regular updates and support from Microsoft, ensuring that the system remains secure and up-to-date with the latest features and improvements. Additionally, it has a user-friendly interface and robust security features, making it suitable for both personal and professional use.

iv. Input Requirements

a. Authentication

Users can input their username, password, or authenticate via their GitHub, Gmail, or Google account. Upon form submission, the system validates the user credentials. For social media accounts, the system checks if the account is already registered.

b. Profile Management

Users will input their information such as complete name, birthday, affiliation, and other information that will be significant to their identification.

c. User Dashboard

For this requirement, the user dashboard should include fields for users to input their search criteria. It may include keywords, filters, or specific parameters related to the datasets they are searching for.

d. Repository Management

In the repository management section, the system should support the uploading of various types of dataset files, including image files (.jpeg, .png, .jpg, webp), tabular data files (.csv), and document files (.docx, .pdf). Additionally, users should be able to upload compressed files (.rar, .7z, .zip) containing datasets. Furthermore, it should allow users to provide detailed

information about the repository, such as a title, description, tags, and other relevant metadata that helps categorizing and identifying datasets.

e. Map Visualization

For map visualization, the system supports GeoJSON files, which contain geographic data that can be used to plot points, lines, and polygons on a map. Additionally, users should be able to input sector details, such as geographical boundaries or regions, to further enhance the map visualization capabilities of the system.

v. Output Requirements

a. Authentication

The output varies depending on the input. If the user credentials are incorrect, an error message is displayed. For social media accounts not registered in the system, the user is redirected to the profile setup page. If the credentials are correct, the user is redirected to the user dashboard.

b. Profile Management

For the output requirements, in profile management, the system should display the updated profile information of the user. This may include their name, contact details, and any other relevant information they have provided. If the form contains any errors, an error message is displayed. If there are no errors, a message is displayed indicating successful profile setup, and the user is redirected to the user dashboard.

c. User Dashboard

In the user dashboard section, the system should present a list of repositories that match the search criteria of the user, allowing them to easily see and access relevant datasets. If the system encounters any issues while fetching the repositories, it will display an error message. If successful, the user will be able to view the list of image repositories.

d. Repository Management

For repository management, the system should show the details of any repositories that the user has created, including the title, description, and tags associated with each repository. It also allows the system to validate the input and update the repository.

e. Map Visualization

In the map visualization section, the system should display a list of the sectors that the user has created. It may provide details such as the sector name and any other relevant information for each sector.

5. Project Management

The proponents will follow the timeline indicated in appendix d figure 14. Following the waterfall model, the Gantt chart was divided into five phases which are the requirements, design, implementation, testing and deployment. Every phase of the project will include proper documentation. The target time frame for this capstone project will be from March 5, 2024 until November 1, 2024.

6. Testing

i. Unit Testing

Individual components of the system will undergo unit testing. This testing process will be carried out by the proponents, the goal is to validate that each unit of the software performs as designed. As the proponents use two different frameworks for backend and frontend, the testing for each component will also utilize a different unit testing framework.

The proponents have chosen storybook for testing the components of the user interface. Storybook allows you to separate a component and document its various use cases in a file with a *.stories.js|ts extension. Regarding the backend functionality, the components will be tested using unittest. Unittest is a Python standard library module which offers a

comprehensive range of tools for creating and executing tests. Altogether, the unit testing will be handled using storybook and unittest.

ii. Integration Testing

The system will undergo incremental integration testing, a method that includes integrating modules individually. It is to ensure that each functions correctly before testing them as a whole system. Storybook for Django emerges as a potent tool for integration testing within Vue.js and Django projects. It is authorized to design and assess UI components in isolation, ensuring compatibility with the Django backend.

Storybook for Django offers several features beneficial for integration testing. It includes middleware that forwards pattern library API requests to Django, enabling testing of components interacting with the Django backend. Being largely framework-agnostic, it integrates smoothly with Storybook's support for HTML and Web Components, including Vue.js. Additionally, it supports automated snapshot testing with Storyshots and allows for automated accessibility and visual regression tests using Jest's jsdom environment. This approach enhances the overall reliability of applications by detecting integration problems at an early stage in the development cycle.

iii. User Acceptance Testing

User acceptance testing will mainly be focused on the end-user’s perspective to the system. The tester for this testing phase will be individuals coming from NOMCAARRD. The proponents will be utilizing the System Usability Scale, a widely used questionnaire-based tool for measuring the perceived usability of a system. The System Usability Scale is a type of Likert Scale consisting of 10 questions that users of your website will be asked to answer.

Appendix E Table 28 provides the questionnaire for determining the SUS. Users will rank each of the 10 template questions on a scale of 1 to 5, indicating their level of agreement. For odd-numbered questions, subtract 1

from the score. For even-numbered questions, subtract the score from 5. Sum these adjusted values to obtain the total score, then multiply by 2.5.

According to Bhat (n.d) if the average System Usability Scale score is 68, scores below this threshold likely indicate significant issues with website usability that need to be addressed. Scores above 68 suggest that the platform's usability surpasses the average.

REFERENCES

Ahmed, A. (2024). Analyzing the Challenges and Opportunities of Knowledge Management in Government Agencies in Egypt. African Journal of Information and Knowledge Management.

Al-Yadumi, S., Tan, E. X., Goh, W. W., & Boursier, P. (2021). Review on Integrating Geospatial Big Datasets. IEEE Access.

Bennet, J. (2008). "Practical Django Projects."Apress

Bhagat, V. (2023). 7 Compelling Reasons Why VueJS is the Perfect Framework for Modern Businesses. PixelCrayons.

Biswas, A. (2023). KRISHI Publication and Data Inventory Repository:An Evaluative Study. ResearchGate.

Bondarenko, E., & Slichna, L. (2023). Development of a Software Application for Geoinformation Mapping of Forest Cover in the Territory of Ukraine Using Remote Sensing Data. International Conference of Young Professionals “GeoTerrace-2023" (pp. 1-5). Lviv: EAGE.

Christensen, J. R., Golden, H. E., Alexander, L., Pickard, B. R., Fritz, K., Lane, C. R., . . & Keefer, M. (2022). Headwater streams and inland wetlands: Status and advancements of geospatial datasets and maps across the United States. Earth-Science Reviews.

Ciolli, G., Meesala, S., Riggs, S. (2019). "PostgreSQL 11 Administration Cookbook."

Djirdeh, H., Lerner, A., Murray, N. (2018). "Fullstack Vue: The Complete Guide to Vue.js and Friends."Fullstack.io

Elmas, F., & Mencütek, Z. (2024). GAPs Data Repository on Return: Guideline, data samples and codebook. Zenodo.

Goteti, G. (2023). Geospatial dataset for Hydrologic analyses in India (GHI): A quality-controlled dataset on river gauges, catchment boundaries and hydrometeorological time series. Earth System Science Data.

Hou, Z., & Pu, Z. (2023). Assessing CYGNSS Satellite Soil Moisture Data for Drought Monitoring with Multiple Datasets and Indicators. Remote Sensing.

Hsu, L., Obe, R. (2015). "PostGIS in Action." Manning Publications.

Innocent, C. (2023). "How to use Axios POST requests." LogRocket.

Ko, G., Lee, J. H., Sim, Y. M., Song, W., Yoon, B.-H., Byeon, I., . . . Hwang, S. (2024). KoNA: Korean Nucleotide Archive as A New Data Repository for Nucleotide Sequence Data. Genomics Proteomics & Bioinformatics.

Kulikov, E., Fedorova, O., Tolmachev, I., Ryazanceva, U., Vrazhnov, D., Gubanov, A., . . . Shmyrina, A. (2023). Russian-language repository of the open clinical data “SibMED Data Clinical Repository”. Bulletin of Siberian Medicine.

Lin, J., Wang, P., Wang, J., Zhou, Y., Zhou, X., Pan, Y., . . . Yang, Z. (2024). An extensive spatiotemporal water quality dataset covering four decades (1980 2022) in China. Earth System Science Data.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., . . . Zitnick, L. (2015). Microsoft COCO: Common Objects in Context. European Conference on Computer Vision.

Liu, J. (2023). Research on Digital Management Construction of Surveying and Mapping Geographic Information Data Archives. Journal of Global Humanities and Social Sciences.

Männikkö, V., Förger, K., Urhonen, H., Tikkanen, J., Antikainen, S., & Munukka, J. (2024). Overview of Finnish national patient data repository for research on medical risk assessment. TechRxiv.

Partarakis, N., Kariyzaki, E., Ntoa, S., Ntagianta, A., Zidianakis, E., & Stephanidis, C. (2023). An Open-data Repository for Sustainable Tourism. Highlights of Sustainability.

Samuel-Rosa, A., Anjos, M. A., Reis, A. M., & Horst, T. Z. (2022). A Repository for Micromorphology Images of Brazilian Soils. I Reunião Brasileira de Micromorfologia de Solos. Piracicaba.

Sicilia, Á., Barriocanal, E., & Alonso, S. (2017). Community curation in open dataset repositories: insights from Zenodo. Procedia Computer Science.

Webster, J. (2016). Data Issues and Promising Practices for Integrated Community Energy Mapping. Natural Resources Canada.

Yuan, Y., Chen, L., Ren, Y., Wang, S., & Li, Y. (2022). Impact of dataset on the study of crop disease image recognition. Int J Agric & Biol Eng.

Zhou, B., Lapedriza, À., Khosla, A., Oliva, A., & Torralba, A. (2017). Places: A 10 Million Image Database for Scene Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Bhat, A. (n.d.). System Usability Scale: What it is, Calculation + Usage. questionpro.

APPENDICES

APPENDIX A

System Analysis Documentation

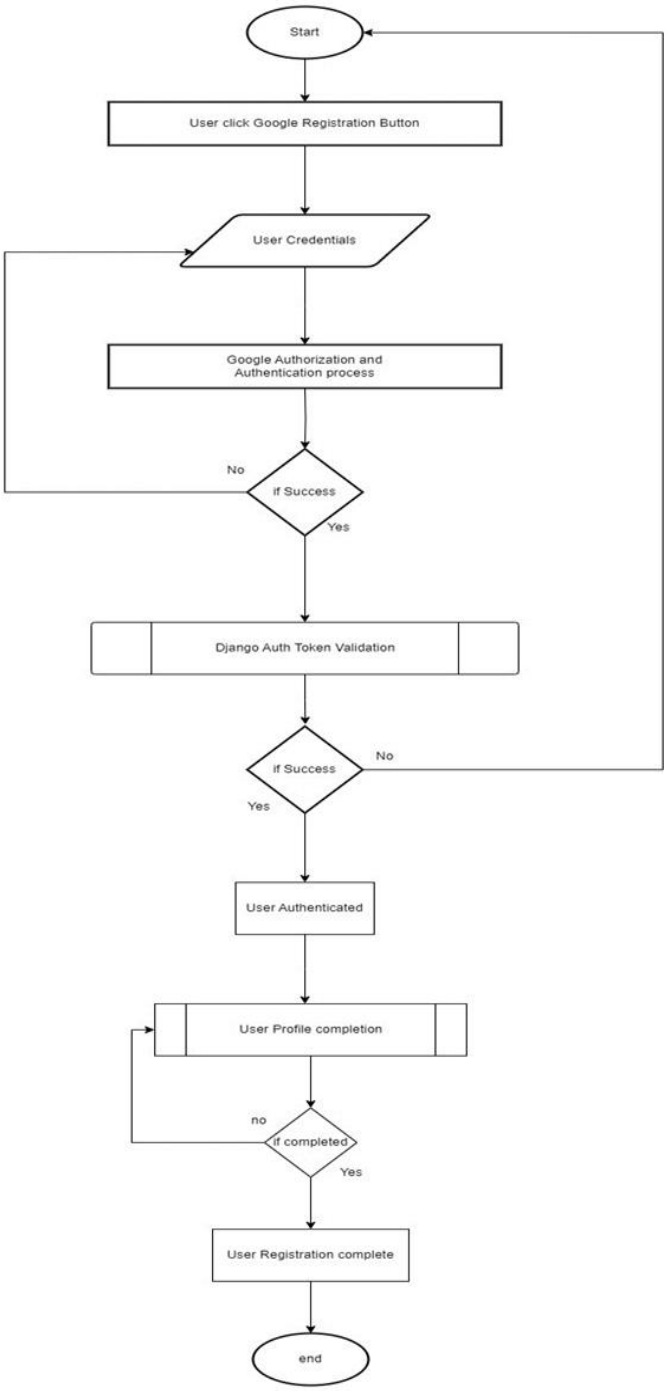

Figure 4. Registration with Google

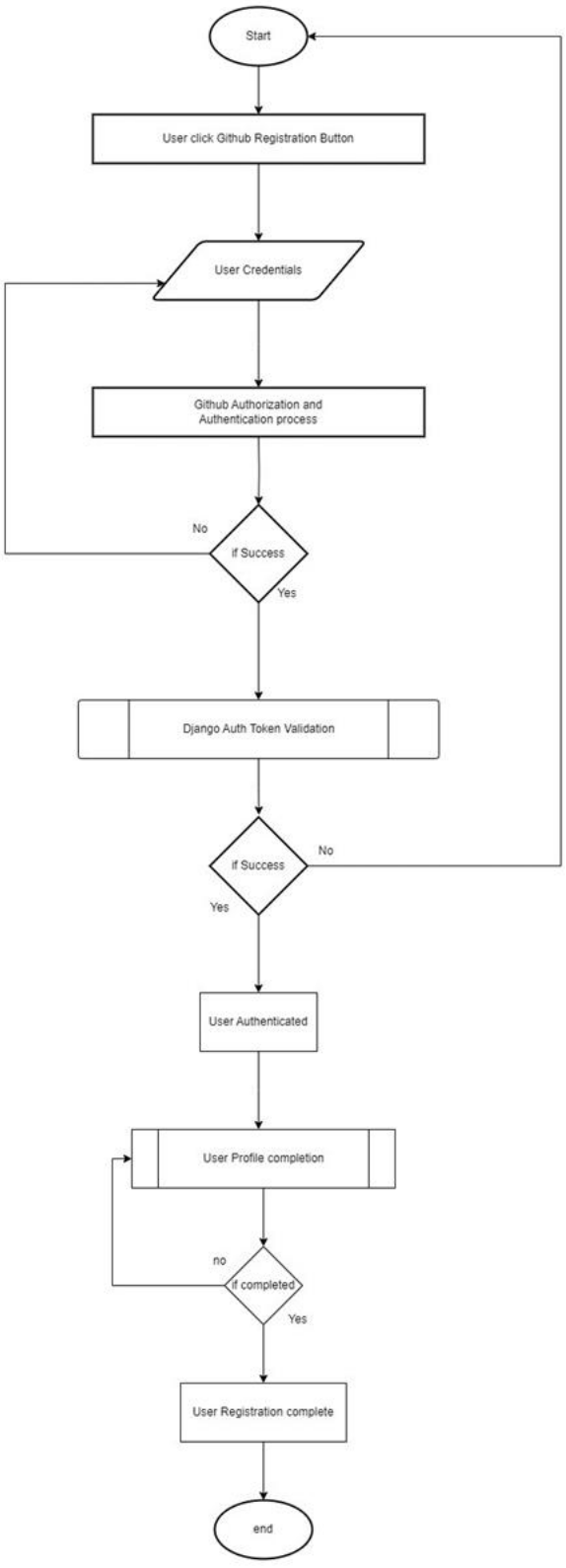

Figure 5. Registration with Github

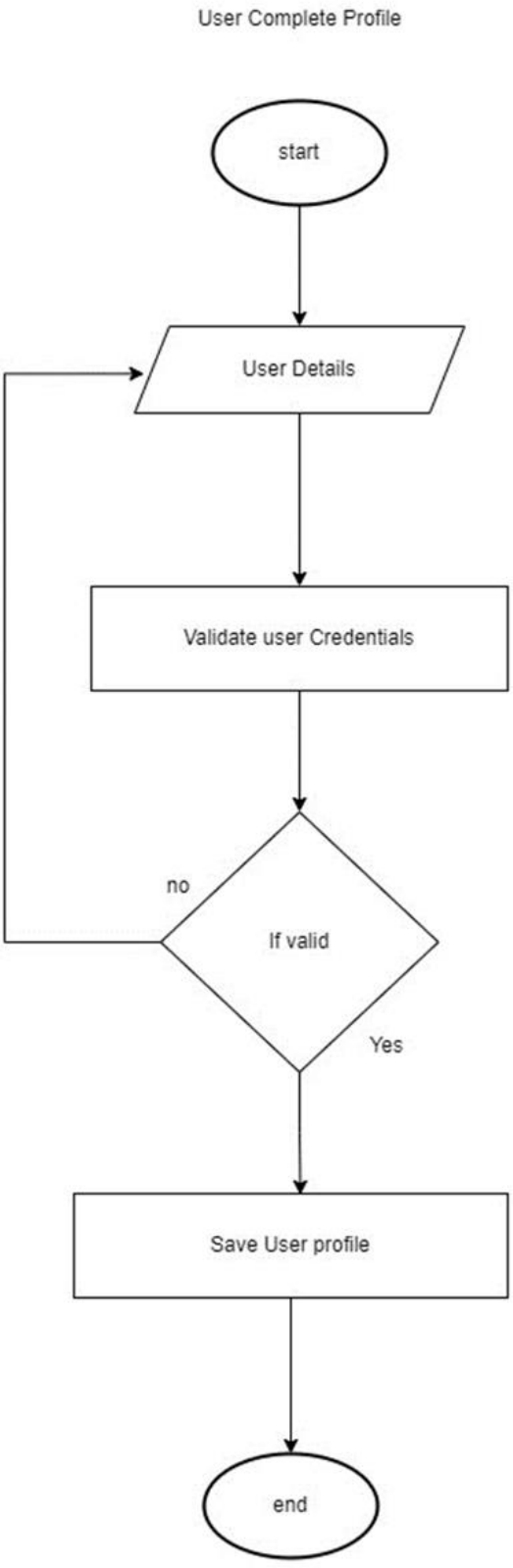

Figure 6. Profile Completion

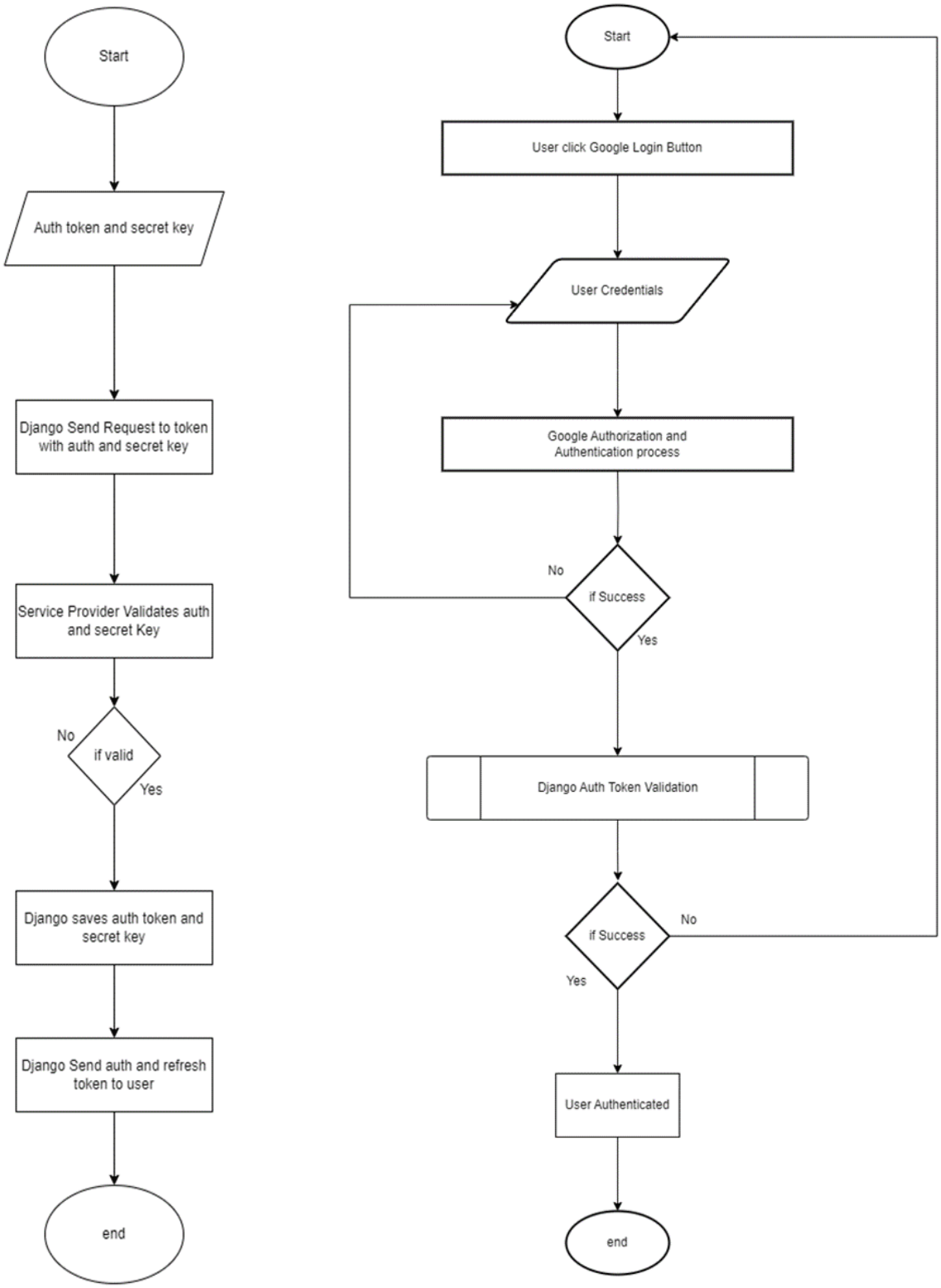

Figure 7. Login through Google

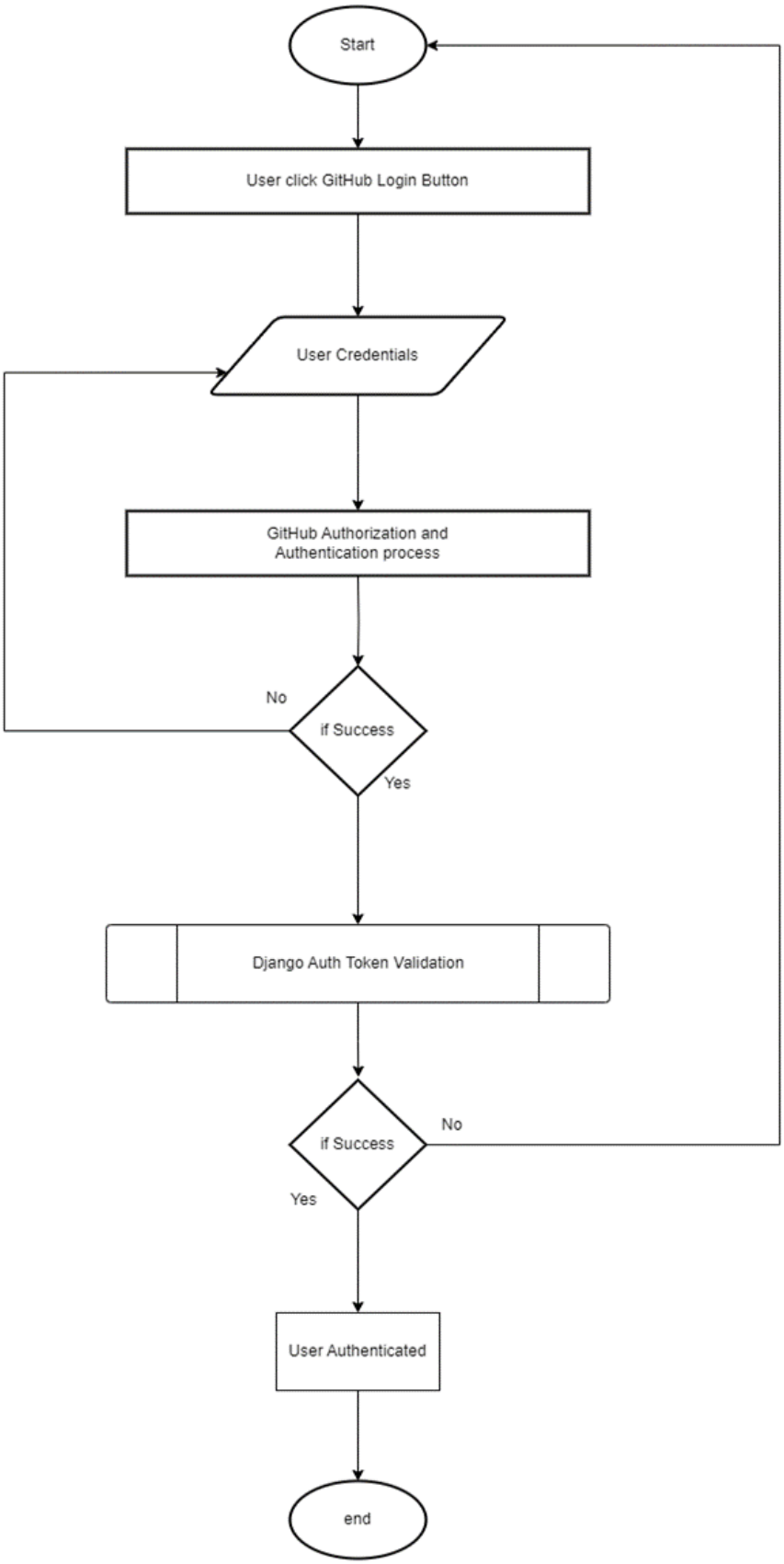

Figure 8. Login through GitHub

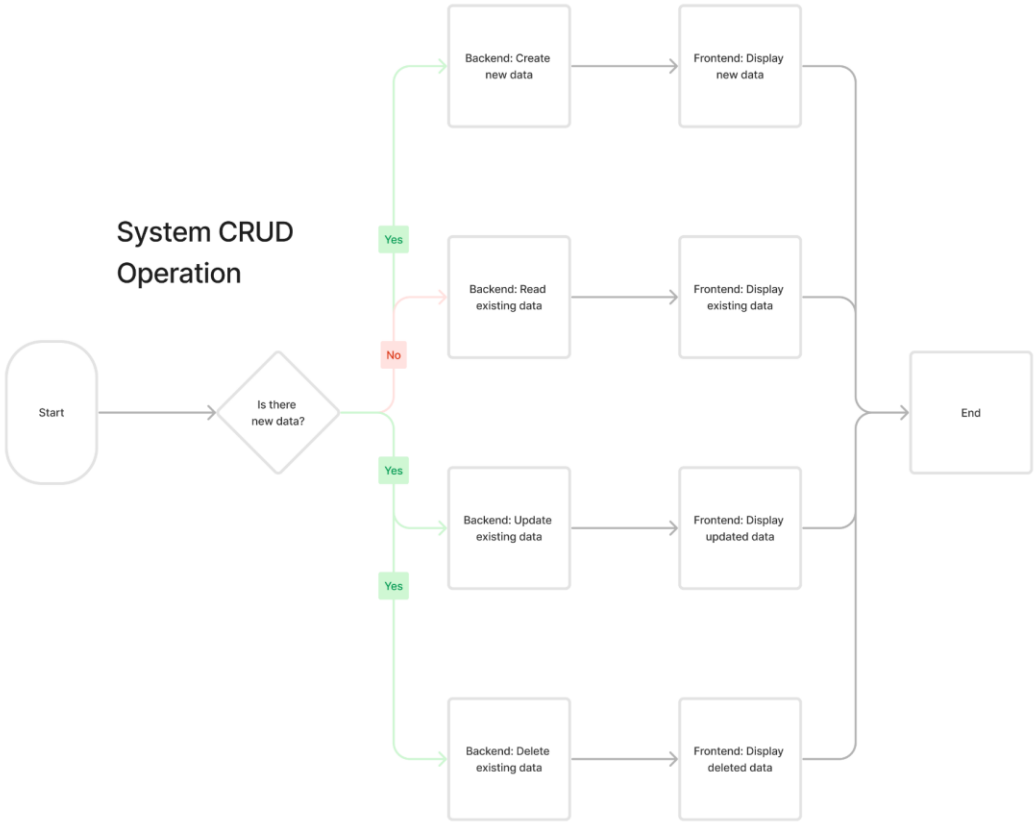

Figure 9. CRUD Data Flowchart

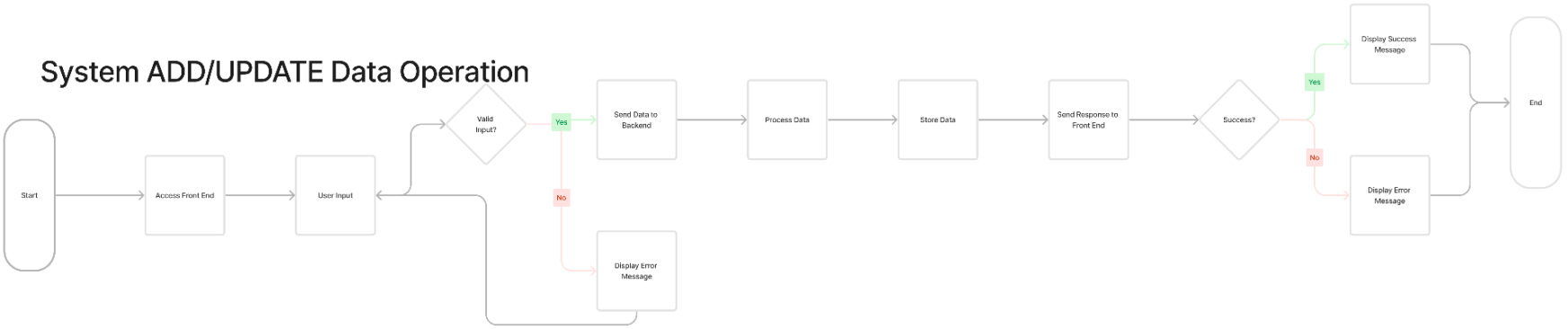

Figure 10. Adding and Update Data Flowchart

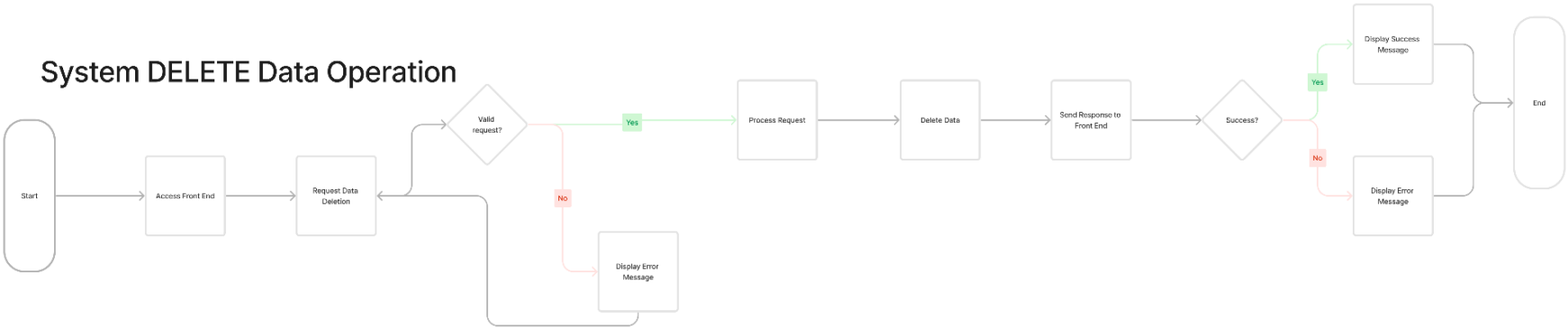

Figure 11. Deleting Data Flowchart

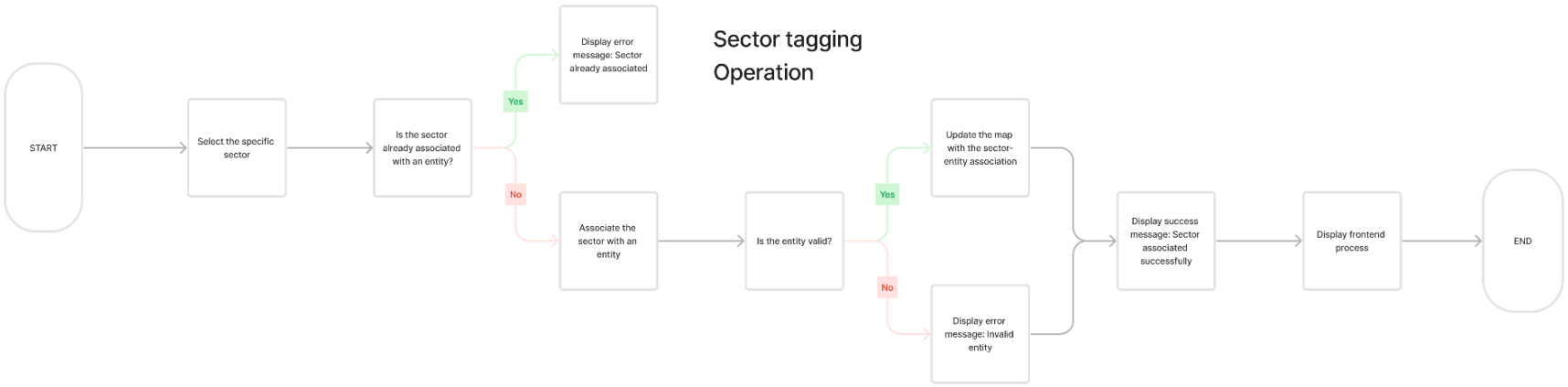

Figure 12. Sector Tagging Flowchart

APPENDIX B

Data Flow Modeling

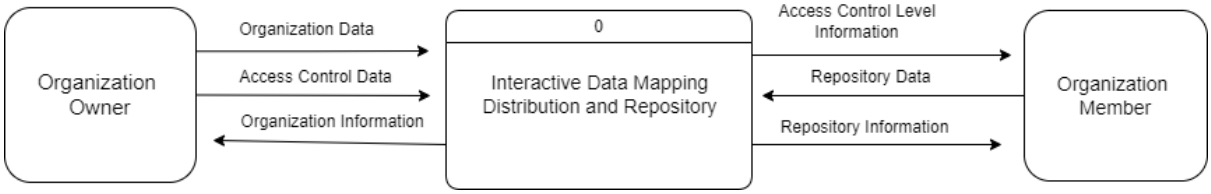

Figure 13. DFD - Context Diagram

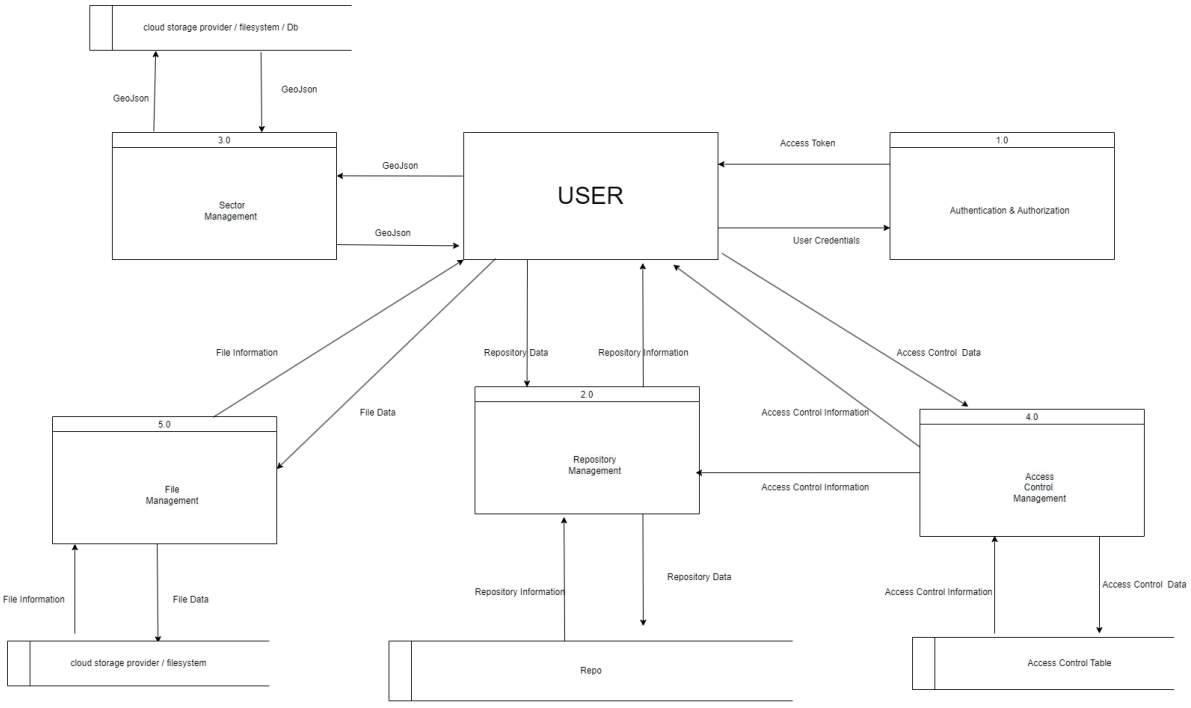

Figure 14. DFD Level-0

APPENDIX C

Use Case Documentation

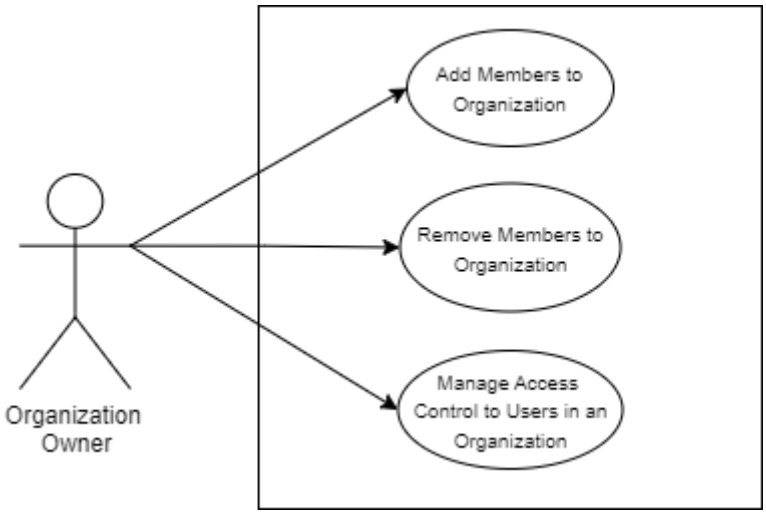

Figure 15. Organization Owner Use Case Diagram

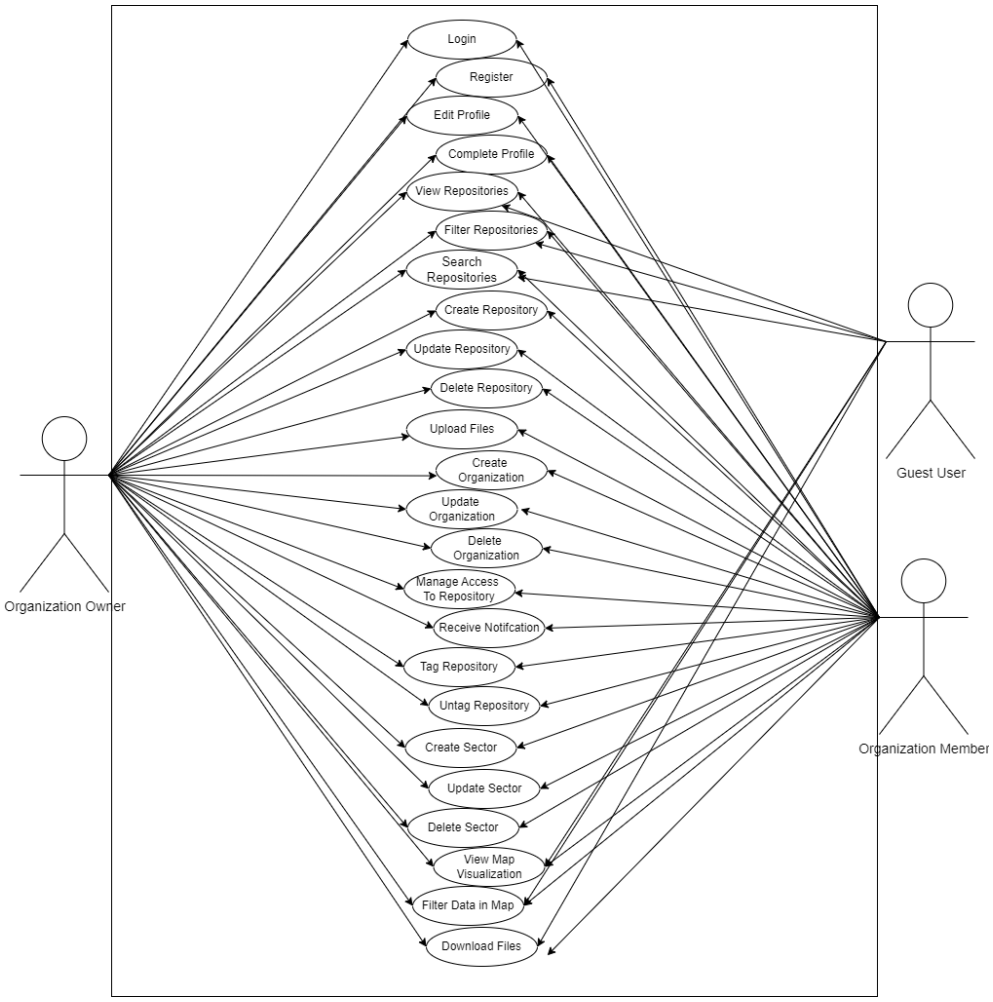

Figure 16. General Use Case Diagram

APPENDIX D

Database Documentation

*

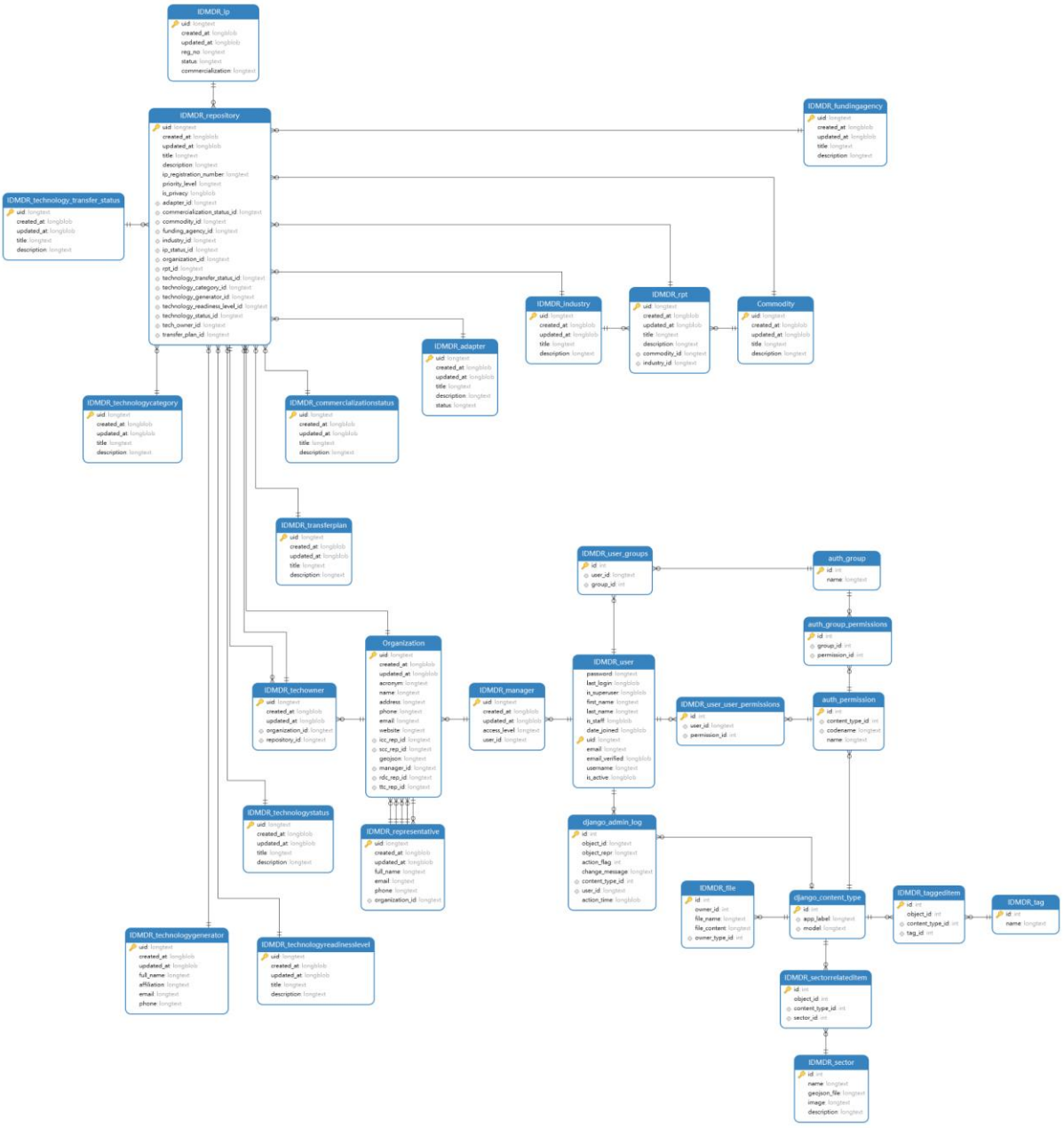

Figure 17. Database Schema for NIDMDR

Table 1. Data Dictionary for Commodity

Field Name	Field Type	Description
Commodity	Model	Represents a commodity.
- title	CharField	Title of the commodity.
- description	TextField	Description of the commodity.
Industry	Model	Represents an industry.
- title	CharField	Title of the industry.
- description	TextField	Description of the industry.
TechnologyCategory	Model	Represents a technology category.
- title	CharField	Title of the technology category.
- description	TextField	Description of the technology category.
TechnologyStatus	Model	Represents a technology status.
- title	CharField	Title of the technology status.
- description	TextField	Description of the technology status.
TechnologyReadinessLevel	Model	Represents a technology readiness level.
- title	CharField	Title of the technology readiness level.
- description	TextField	Description of the technology readiness level.
RPT	Model	Represents a research project conducted for the technology.
- title	CharField	Title of the research project.
- description	TextField	Description of the research project.
- commodity	ForeignKey(Commodity)	Foreign key relationship with Commodity model.

- industry	ForeignKey(Industry)	Foreign key relationship with Industry model.
FundingAgency	Model	Represents a funding agency.
- title	CharField	Title of the funding agency.
- description	TextField	Description of the funding agency.
technologyGenerator	Model	Represents a technology generator.
- full_name	CharField	Full name of the technology generator.
- affiliation	CharField	Affiliation of the technology generator.
- email	EmailField	Email address of the technology generator.
- phone	CharField	Phone number of the technology generator.
TechOwner	Model	Represents a technology owner.
- organization	ForeignKey(Organization)	Foreign key relationship with Organization model.
- repository	ForeignKey(Repository)	Foreign key relationship with Repository model.
TransferPlan	Model	Represents a transfer plan.
- title	CharField	Title of the transfer plan.
- description	TextField	Description of the transfer plan.
Adapter	Model	Represents an adapter.
- title	CharField	Title of the adapter.
- description	TextField	Description of the adapter.
- status	CharField	Status of the adapter.
Technology_Transfer_Status	Model	Represents a technology transfer status.

- title	CharField	Title of the technology transfer status.
- description	TextField	Description of the technology transfer status.
IP	Model	Represents an intellectual property status.
- reg_no	CharField	Registration number of the intellectual property.
- status	CharField	Status of the intellectual property.
- commercialization	TextField	Commercialization status of the intellectual property.
CommercializationStatus	Model	Represents a commercialization status.
- title	CharField	Title of the commercialization status.
- description	TextField	Description of the commercialization status.
Repository	Model	Represents a repository.
- title	CharField	Title of the repository.
- description	TextField	Description of the repository.
- ip_registration_number	CharField	Registration number of the intellectual property associated with the repository.
- priority_level	CharField	Priority level of the repository.
- is_privacy	BooleanField	Indicates if the repository is private or not.

Table 2. Data Dictionary for Repository

Field Name	Field Type	Description
Repository (continued)		
- commodity	ForeignKey(Commodity)	Foreign key relationship with Commodity model.
- industry	ForeignKey(Industry)	Foreign key relationship with Industry model.
- technology_category	ForeignKey(TechnologyCategory)	Foreign key relationship with TechnologyCategory model.
- technology_status	ForeignKey(TechnologyStatus)	Foreign key relationship with TechnologyStatus model.
- technology_readiness_level	ForeignKey(TechnologyReadinessLevel)	Foreign key relationship with TechnologyReadinessLevel model.
- rpt	ForeignKey(RPT)	Foreign key relationship with RPT model.
- funding_agency	ForeignKey(FundingAgency)	Foreign key relationship with FundingAgency model.
- technology_generator	ForeignKey(technologyGenerator)	Foreign key relationship with technologyGenerator model.
- organization	ForeignKey(Organization)	Foreign key relationship with Organization model.
- tech_owner	ForeignKey(TechOwner)	Foreign key relationship with TechOwner model.
- transfer_plan	ForeignKey(TransferPlan)	Foreign key relationship with TransferPlan model.
- adapter	ForeignKey(Adapter)	Foreign key relationship with Adapter model.

- technology_transfer_status	ForeignKey(Technology_Transfer_Status)	Foreign key relationship with Technology_Transfer_Status model.
- ip_status	ForeignKey(IP)	Foreign key relationship with IP model.
- commercialization_status	ForeignKey(CommercializationStatus)	Foreign key relationship with CommercializationStatus model.
ip_registration_number	Varchar(100)	Ip registration Number
priority_level	Varchar(100)	Priority level with defined choices
Is_private	Bool	Define the privacy of the a repository

Table 3. Data Dictionary for Manager

Field Name	Field Type	Description
Manager	Model	Represents a manager.
- user	OneToOneField(User)	One-to-one relationship with the User model.
- access_level	CharField	Access level of the manager. Choices: 'admin', 'manager', 'viewer'. Default is 'viewer'.
Representative	Model	Represents a representative.
- full_name	CharField	Full name of the representative.
- email	EmailField	Email address of the representative.
- phone	CharField	Phone number of the representative.
- organization	ForeignKey(Organization)	Foreign key relationship with Organization model.
Organization	Model	Represents an organization.
- acronym	CharField	Acronym of the organization.
- name	CharField	Name of the organization.

- address	TextField	Address of the organization.
- manager	ForeignKey(Manager)	Foreign key relationship with Manager model.
- phone	CharField	Phone number of the organization.
- email	EmailField	Email address of the organization.
- website	URLField	Website URL of the organization.
- ttc_rep	ForeignKey(Representative)	Foreign key relationship with Representative model for technology transfer cluster representative.
- icc_rep	ForeignKey(Representative)	Foreign key relationship with Representative model for information communication cluster representative.
- scc_rep	ForeignKey(Representative)	Foreign key relationship with Representative model for science communication cluster representative.
- rdc_rep	ForeignKey(Representative)	Foreign key relationship with Representative model for research development cluster representative.
- geojson	JSONField	GeoJSON data of the organization.

Table 4. Data Dictionary for File

Field Name	Field Type	Description
File	Model	Represents a file.
- owner_type	ForeignKey(ContentType)	Foreign key to ContentType model.
- owner_id	PositiveIntegerField()	ID of the owner.
- owner	GenericForeignKey('owner_type', 'owner_id')	Generic foreign key representing the owner of the file.
- file_name	CharField(max_length=255)	Name of the file.
- file_content	FileField(upload_to='files/')	Field to store the actual file content.
Sector	Model	Represents a sector.

- name	CharField(max_length=255)	Name of the sector.
- geojson_file	FileField(upload_to='geojson_files/')	Field to store the GeoJSON file related to the sector.
- image	ImageField(upload_to='sector_images/')	Field to store the image related to the sector.
- description	TextField()	Description of the sector.
SectorRelatedItem	Model	Represents a related item for a sector.
- sector	ForeignKey(Sector)	Foreign key relationship with Sector model.
- content_type	ForeignKey(ContentType)	Foreign key to ContentType model.
- object_id	PositiveIntegerField()	ID of the related object.
- content_object	GenericForeignKey('content_type', 'object_id')	Generic foreign key representing the related object.

Table 5. Data Dictionary for Tag

Field Name	Field Type	Description
Tag	Model	Represents a tag.
- name	CharField(max_length=50)	Name of the tag.
TaggedItem	Model	Represents a tagged item.
- tag	ForeignKey(Tag)	Foreign key relationship with Tag model.
- content_type	ForeignKey(ContentType)	Foreign key to ContentType model.
- object_id	PositiveIntegerField()	ID of the related object.
- content_object	GenericForeignKey('content_type', 'object_id')	Generic foreign key representing the related object.

Table 6. Data Dictionary for User

Field Name	Field Type	Description
User	Model (Custom)	Represents a user.
- uid	UUIDField(primary_key=True)	Primary key field representing the UUID of the user.
- password	CharField	Field to store the password of the user.
- last_login	DateTimeField	Date and time when the user last logged in.
- is_superuser	BooleanField(default=False)	Indicates whether the user is a superuser or not.
- username	CharField(max_length=150, blank=True, null=True)	Username of the user.
- first_name	CharField(max_length=30, blank=True)	First name of the user.
- last_name	CharField(max_length=150, blank=True)	Last name of the user.
- email	EmailField(unique=True)	Email address of the user.
- email_verified	BooleanField(default=False)	Indicates whether the user's email has been verified or not.
- is_staff	BooleanField(default=False)	Indicates whether the user is staff or not.
- is_active	BooleanField(default=False)	Indicates whether the user account is active or not.
- date_joined	DateTimeField	Date and time when the user account was created.

APPENDIX D

System Functional Requirement Specification

Table 7. Login Requirement Specification

Requirement No. 1	**Login**
Priority	High
Purpose	To enable users to access the system through their account credentials or social media accounts.
Input	Users can input their username, password, or authenticate via their GitHub, Gmail, or Google account.
Operations	Upon form submission, the system validates the user credentials. For social media accounts, the system checks if the account is already registered.
Output	The output varies depending on the input. If the user credentials are incorrect, an error message is displayed. For social media accounts not registered in the system, the user is redirected to the profile setup page. If the credentials are correct, the user is redirected to the user dashboard.

Table 8. Registration Requirement Specification

Requirement No. 2	**Register**
Priority	High
Purpose	To enable users to register to the system using their account credentials or social media accounts.
Input	Users can input their username, password, and name or authenticate via their GitHub, Gmail, or Google account.
Operations	Upon form submission, the system validates the user credentials. For social media accounts, the system checks if the account is already registered.
Output	The output varies depending on the input. If the username or password already exists, an error message is displayed. If the social media account is already registered in the system, the user is redirected to the user dashboard. If the user credentials return no error, the user is redirected to the profile setup page.

Table 9. Profile Completion Requirement Specification

Requirement No. 3	Complete Profile
Priority	High
Purpose	To allow users to set up their profile information after registration, which will be displayed in the system.
Input	Users will input their information such as complete name, birthday, etc.
Operations	Upon form submission, the system validates the profile information.
Output	If the form contains any errors, an error message is displayed. If there are no errors, a message is displayed indicating successful profile setup, and the user is redirected to the user dashboard.

Table 10. Profile Management Requirement Specification

Requirement No. 4	Edit Profile
Priority	Medium
Purpose	To allow users to edit their profile information.
Input	Users will input their information such as complete name, birthday, etc.
Operations	Upon form submission, the system validates the profile information.
Output	If the form contains any errors, an error message is displayed. If there are no errors, a message is displayed indicating the profile information has been successfully updated.

Table 11. View Repository Requirement Specification

Requirement No. 5	View Repositories
Priority	Medium
Purpose	To allow users to view data repositories
Input	Users will not input any information. The system will display the repositories available.
Operations	The user will view the list of repositories. The system will fetch and display the repositories.
Output	If the system encounters any issues while fetching the repositories, it will display an error message. If successful, the user will be able to view the list of image repositories.

Table 12.Repository Filtering Requirement Specification

Requirement No. 6	Filter Repositories
Priority	Medium
Purpose	To allow users to filter repositories
Input	Users will select an option on how they would like to filter the list of repositories.
Operations	The user will select an option to filter the repositories. The system will fetch and display the repositories based on the user's filter.
Output	If the system encounters any issues while fetching the repositories, it will display an error message. If successful, the user will be able to view the filtered list of repositories.

Table 13. Repository Searching Requirement Specification

Requirement No. 7	Search Repositories
Priority	Medium
Purpose	To allow users to search repositories
Input	Users will input a search term
Operations	The user will input a string and search the repositories. The system will fetch and display the repositories that match the search term

Output	If the system encounters any issues while fetching the repositories, it will display an error message. If there are no available results based on the search query of the user, it will display a "not found" message. If successful, the user will be able to view the list of image repositories that match the search query.

Table 14. Repository Creation Requirement Specification

Requirement No. 8	**Create Repository**
Priority	High
Purpose	To allow users to create repositories
Input	Users will input the details in creating repositories such as repository name and the files
Operations	The user will input the required details and submit the form to create a new repository. The system will validate the input and create the repository.
Output	If the system encounters any issues while creating the repository, it will display an error message. If the repository is created successfully, it will display a confirmation message and the user will be redirected to the new repository.

Table 15. Repository Modification Requirement Specification

Requirement No. 9	**Edit Repository**
Priority	Medium
Purpose	To allow users to edit existing repositories
Input	Users will input the details in the fields they want to edit in the repositories such as repository name and the repository files.
Operations	The user will input the required details and submit the form to edit the repository. The system will validate the input and update the repository.

Output	If the system encounters any issues while creating the repository, it will display an error message. If the repository is updated successfully, it will display a confirmation message and the user will be redirected to the updated repository.

Table 16. Repository Deletion Requirement Specification

Requirement No. 10	**Delete Repository**
Priority	High
Purpose	To allow users to delete existing repositories.
Input	Users will not input anything but will select the repository they want to delete.
Operations	The user will select the repository they want to delete. The system will send a confirmation dialog.
Output	The system will ask for confirmation before deletion. If it is cancelled, then it will close the dialog. If the user confirmed the deletion, it will display a message is deleted successfully then the user will be redirected to the user's repositories dashboard

Table 17. Repository Insertion Requirement Specification

Requirement No. 11	**Add Files to the Repository**
Priority	High
Purpose	To allow users to add files in an existing repository
Input	Users will upload individual files or zip containing the necessary data.
Operations	The user will upload the files. The system will validate the files
Output	If the system encounters any issues while validating the files such as invalid files, it will display an error message. If the data are uploaded successfully, it will display a message and the user will be redirected to the repository.

Table 18. Repository Availability Requirement Specification

Requirement No. 12	**Change Repository Availability**
Priority	Low
Purpose	To allow users to change the publicity of their repositories (public, private)
Input	Users will select if they want to make their repository public or private
Operations	The user will select the visibility option for the repository. The system will validate the input and update the repository's visibility.
Output	If the system encounters issues while updating the repository's visibility, it will display an error message. If the repository's visibility is updated successfully, it will display a confirmation message while the user will be redirected to the repository. The access level will be changed to either public or private considering the user input.

Table 19. System NotificationRequirement Specification

Requirement No. 13	**Receive Notifications**
Priority	Medium
Purpose	Enable users to receive notifications
Input	Users will not need to input anything.
Operations	The system will wait for any event that will trigger to send a notification.
Output	The user will receive a notification with details of what the notification is about. If the user clicks the specific notification, the user will be redirected to the page with what the notification is about. (e.g. If the user receives a notification that their request access is granted, the user will be redirected to that repository.)

Table 20. Repository Accessibility Requirement Specification

Requirement No. 14	**Manage Access to Repository**
Priority	Medium
Purpose	Allow users to approve access or decline access requests to their repository.
Input	Users will select either to approve or decline the request
Operations	The user will select either to approve or delete the request and the system will send the message to the user who sent the request
Output	If the system encounters any issues while approving or declining a request, it will display an error message. If the request is successful, it will return a message.

Table 21. System Organization's Creation Requirement Specification

Requirement No. 15	**Create Organization**
Priority	High
Purpose	Allow users to create organization
Input	Users will input necessary data in creating the organization such as organization name, description, etc.
Operations	The user will input the required details and submit the form to create the organization. The system will validate the input.
Output	If the system encounters any issues while creating the organization, it will display an error message. If successful, the user will be redirected to the created organization

Table 22. Organization Modification Requirement Specification

Requirement No. 16	**Update Organization**
Priority	Medium
Purpose	Allow users to edit and update existing organization
Input	Users will input necessary data in the fields they want to edit in the organization

Operations	The user will input the required details and submit the form to update the organization. The system will validate the input.
Output	If the system encounters any issues while updating the organization, it will display an error message. If successful, the user will be redirected to the created organization

Table 23. Organization Deletion Requirement Specification

Requirement No. 17	**Delete Organization**
Priority	Low
Purpose	Allow users to delete existing organizations.
Input	Users will not input anything but will select the organization they want to delete.
Operations	The user will select the organization they want to delete. The system will send a confirmation dialog.
Output	The system will ask for confirmation before deletion. If it is cancelled, then it will close the dialog. If the user confirms the deletion, it will display a message that it is deleted successfully then the user will be redirected to the list of organizations.

Table 24. Organization Insertion Requirement Specification

Requirement No. 18	**Add Members to Organization**
Priority	High
Purpose	Allow organization owner to add members
Input	Owner will not input anything but will select the existing users they want to add to the organization
Operations	The user will select the users they want to add. The system will send a confirmation dialog.
Output	If the system encounters any issues while adding users, it will display an error message. If successful, the owner will

	be redirected to the page which shows the list of the members of the organization

Table 25. Organization Member Exclusion Requirement Specification

Requirement No. 19	**Remove Members to Organization**
Priority	Low
Purpose	To allow the organization owner to remove unwanted members to the organization.
Input	Owner will not input anything but will select the existing users they want to remove to the organization.
Operations	The user will select the users they want to remove. The system will send a confirmation dialog.
Output	If the system encounters any issues while removing the users, it will display an error message. If successful, the owner will be redirected to the page which shows the list of the members of the organization

Table 26. Organization Accessibility Management Requirement Specification

Requirement No. 20	**Manage Access Control to Users**
Priority	High
Purpose	Allow the organization owner to give certain access control to the members of the association.
Input	The owner will select a member and select the desired access control for that person.
Operations	The owner will select a member and select the desired access control for that person. The system will send a confirmation dialog.

Output	If the system encounters any problems while processing the request, it will respond with an error message. If successful, the owner will be redirected to the managed access control settings page of the organization with the updated access control for the selected member.

Table 27. Sector Creation Requirement Specification

Requirement No. 21	**Create a Sector**
Priority	High
Purpose	Grant users to create a sector on the map.
Input	User will input the coordinates of the map.
Operations	The system will prompt the user if the user will import an existing geoJSON or to draw a polygon from the map.
Output	If the system encounters any issues while importing the file or while creating the polygon, it will display an error message. If successful, the user will proceed to tag a repository with the created sector.

Table 28. Sector Modification Requirement Specification

Requirement No. 22	**Update Sector**
Priority	Medium
Purpose	Enable users to edit and update existing sector
Input	Users will choose the sector they wish to modify and fill in the fields that they want to change.
Operations	The user will choose the sector they wish to modify, input the necessary details, and submit the form to modify it. The system will validate the input and update the post
Output	If the system encounters any issues while updating the chosen campaign, it will display an error message. If the update is successful, it will display a confirmation message and redirect the user to the map sector page.

Table 29. Sector Deletion Requirement Specification

Requirement No. 23	**Delete Sector**
Priority	Low
Purpose	To allow users to delete existing sectors.
Input	Users will not need to input anything but will choose the sectors they want to remove.
Operations	The user will select the sector they want to delete. The system will send a confirmation dialog.
Output	The system will ask for confirmation before removal. If it is cancelled, then it will close the dialog. If the user confirms the removal, it will display a message that the sector has been removed successfully then redirect the user to the sector page.

Table 30. Repository Tagging Requirement Specification

Requirement No. 24	**Tagging of Repository**
Priority	Medium
Purpose	To allow users to associate a repository with a sector.
Input	Users will input the required data for repository tagging and select the repository they wish to link with the sector.
Operations	There are two methods for tagging a repository. First, users can create a sector and then tag a repository to it. Alternatively, users can find an existing sector, select it, and click the "Tag Repository with this Sector" button to input the necessary data. The system will then validate the input.
Output	If the system encounters any issues while processing the request, it will display an error message. If the request is successful, it will display a success message and redirect the user to the rewards page.

Table 31. Repository Untagging Requirement Specification

Requirement No. 25	Untagging of Repository
Priority	Low
Purpose	To allow users to disassociate a repository from a sector.
Input	Users will select a sector and then choose the repository they wish to untag.
Operations	Users can untag a repository by either selecting an existing sector and clicking the "Untag Repository from this Sector" button or by directly selecting the repository they wish to untag from a list of tagged repositories. The system will then process the request to remove the association.
Output	If the system encounters any issues while processing the request, it will display an error message. If the request is successful, it will display a success message and redirect the user to the sector's page where the repository was previously tagged.

Table 32. Map Visualization Requirement Specification

Requirement No. 26	View Map Visualization
Priority	High
Purpose	Enable users to view the Map
Input	Users will not need to input anything.
Operations	The user will select the navigation item for the Map
Output	After selecting the navigation link, the user will be redirected to the map visualization page.

Table 33. Map Data Filtering Requirement Specification

Requirement No. 27	Filtering of Data in Map
Priority	Low
Purpose	To allow users to filter the data displayed on the map.
Input	Users will select a specific sector and choose from the available filters.
Operations	Users can apply filters to the data displayed on the map by selecting a sector and then choosing the filters they wish to apply. The system will then process the request to filter the data based on the user's selection.
Output	If the system encounters any issues while processing the request, it will display an error message. If the request is successful, it will display the updated map with the filtered data.

Table 34. Repository Files Downloading Requirement Specification

Requirement No. 27	Downloading of Files in Repository
Priority	High
Purpose	Enable users to download files from a specific repository.
Input	No user input required.
Operations	Users must select the file they wish to download. Alternatively, they can opt to download all files. Upon clicking the download button, the selected files will be downloaded.
Output	If any issues arise during processing, the system will show an error message. Upon successful completion, the downloaded files will be saved to the user's file storage. If the user chooses to download all files in the repository, they will be saved as a zip file.

APPENDIX E

Project Management Documentation

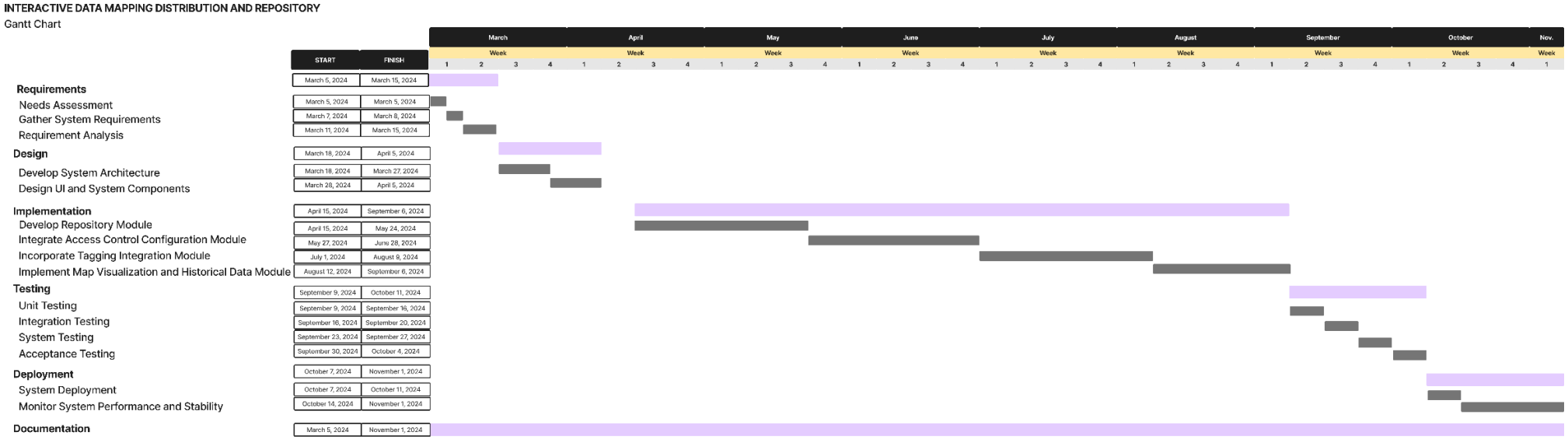

Figure 18. Gantt Chart for NIDMDR

APPENDIX E

User Acceptance Testing

Table 35. System Usability Scale Questionnaire

QUESTIONS	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
	(1)	(2)	(3)	(4)	(5)
1. I think that I would like to use this system frequently.					
2. I found the system unnecessarily complex.					
3. I thought the system was easy to use.					
4. I think that I would need the support of a technical person to be able to use this system.					
5. I found the various functions in this system were well integrated.					
6. I thought there was too much inconsistency in this system.					
7. I would imagine that most people would learn to use this system very quickly.					
8. I found the system very cumbersome to use.					
9. I felt very confident using the system.					
10. I needed to learn a lot of things before I could get going with this system.					