Nature's Palette: An open-access digital repository of spectral data

Comp 6905-001 – Software Engineering

Final Project

Fall 2019

By

Sourav Barua	[201991158]
Vivekkumar Dharmendrakumar Patel	[201991314]
Lavaanantha Thayaparendran	[201990251]
Shiplu Das	[201891201]
Sourav Sarker	[201892985]

Table of Contents

01. Introduction 3
02. System Model 4
02.1. Use Case Diagrams and Descriptions 4
2.2. Design Goals 15
2.3. Logical Architecture 17
03. User Manual (Readme) 18
04. Web Link 20

01. Introduction

In the world internet technology, it is now very flexible for anyone to access any information they need. Just a few clicks a user need to open the world of information. However, the question arises are all the information we get over are legitimate? The word “free” give access to many fake users to upload and mis-lead people. Dealing with wrong information may not only be the reason of losing someone’s motivation over the work, it might also cause a violation of the society or even worse. For someone doing research on a topic, it is always important to assure that the data came from an authentic source.

The scope of the Nature’s Palette system is to develop a digital repository to allow researcher all around the world to upload their analysis data (spectrometer reading) in a simple way. So that the other people who needs those data can access this repository system and search for what they want and can download in in simple steps. The system was designed to allow only authenticated researchers to upload the data. For that it was planned to intergrade the authentication with ORCID.

The data can be large set of files (readings) with raw data and metadata. Obviously, there can be errors in files. So that the validation step was implemented in upload process to check all the files before pushing it to the repository database. Once it’s uploaded, the data will be cleaned by the given script of filtering mechanism and matrix file will be created. After that all the files should be stored in backend database.

Any interested user can access the repositories (as a guest) and search for any required data and will be able to download raw data and metadata file. Searching will be based on the metadata file column descriptions and has the facility to use AND OR logics in searching data.

02. System Model

02.1. Use Case Diagrams and Descriptions

Use Case Diagram

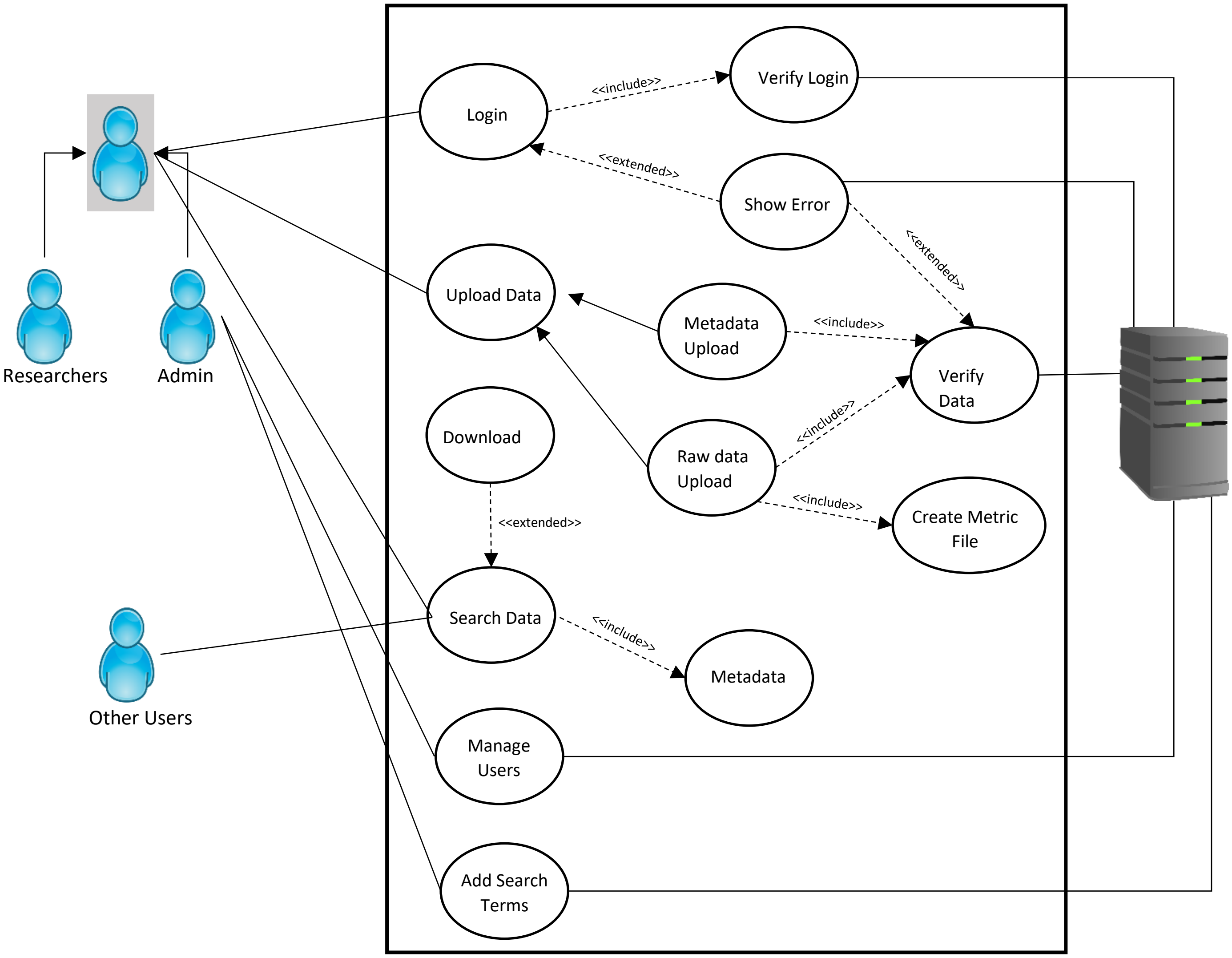

Fig 2.1.1: Use case diagram

File Upload - Diagram

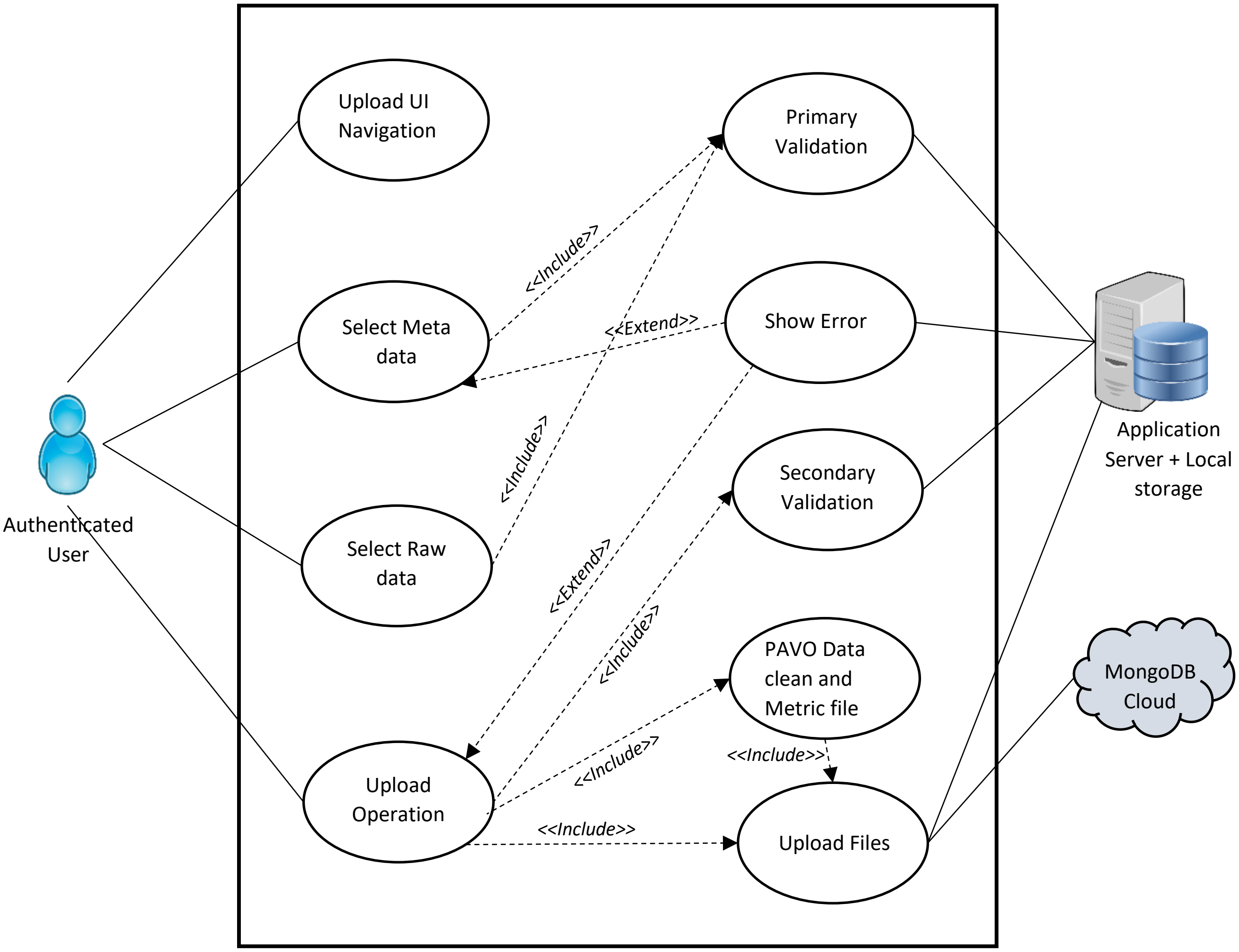

Fig 2.1.2: File upload Use case diagram

File Upload – Textual Description

Name:	Upload Raw data & Metadata files
Participating Actor:	Researcher (Authenticated Users)
Entry Condition:	Researcher should be a registered user
Flow of Events:	1. Researcher logged into the system and press the upload button. 2. System redirect the user to upload instruction page and ask the user for acknowledgement. 3. Researcher read the instruction and agrees with the terms and condition. Also, read and accept the format at which the files need to be maintained. 4. System shows the user the basic form for upload. 5. Researcher fill the form and click on Next to proceed. 6. System checks whether all the mandatory fields are completed and proceed to the next step of uploading. 7. Researcher upload the raw data and metadata files and click on submit button. 8. System verifies the file and show the user a successful message that the file has been successfully uploaded and will be notified when the data will be released for download. 9. System will perform the raw data validation (No reading should be less than -2) and notify user if its clean and push to database. 10. System computes the metrics for uploaded raw files and store it to the repositories.
Exit Condition:	Files will be added to the repositories and will be available for other users to search and

	download.
Alternate Flows:	Researcher enters incorrect login credentials. a. System show error message. b. User reads the error message. c. System redirect the user to login page and ask the user to enter correct credentials again. d. Researcher enters correct credentials and click on the Login button. 2. Researcher uploaded the raw file or metadata file in the wrong format. a. System shows an error to the user. b. Researcher read the error message c. System redirect the user to the upload page d. Researcher correct the format and upload it again e. System shows success message. 3.Raw data file contains negative values. a) If there any values between 0 to -2, notify user with warning and accept the file b) If there any values less than -2, notify user with error and drop that raw file and relevant metadata entry.

File Modify Request - Diagram

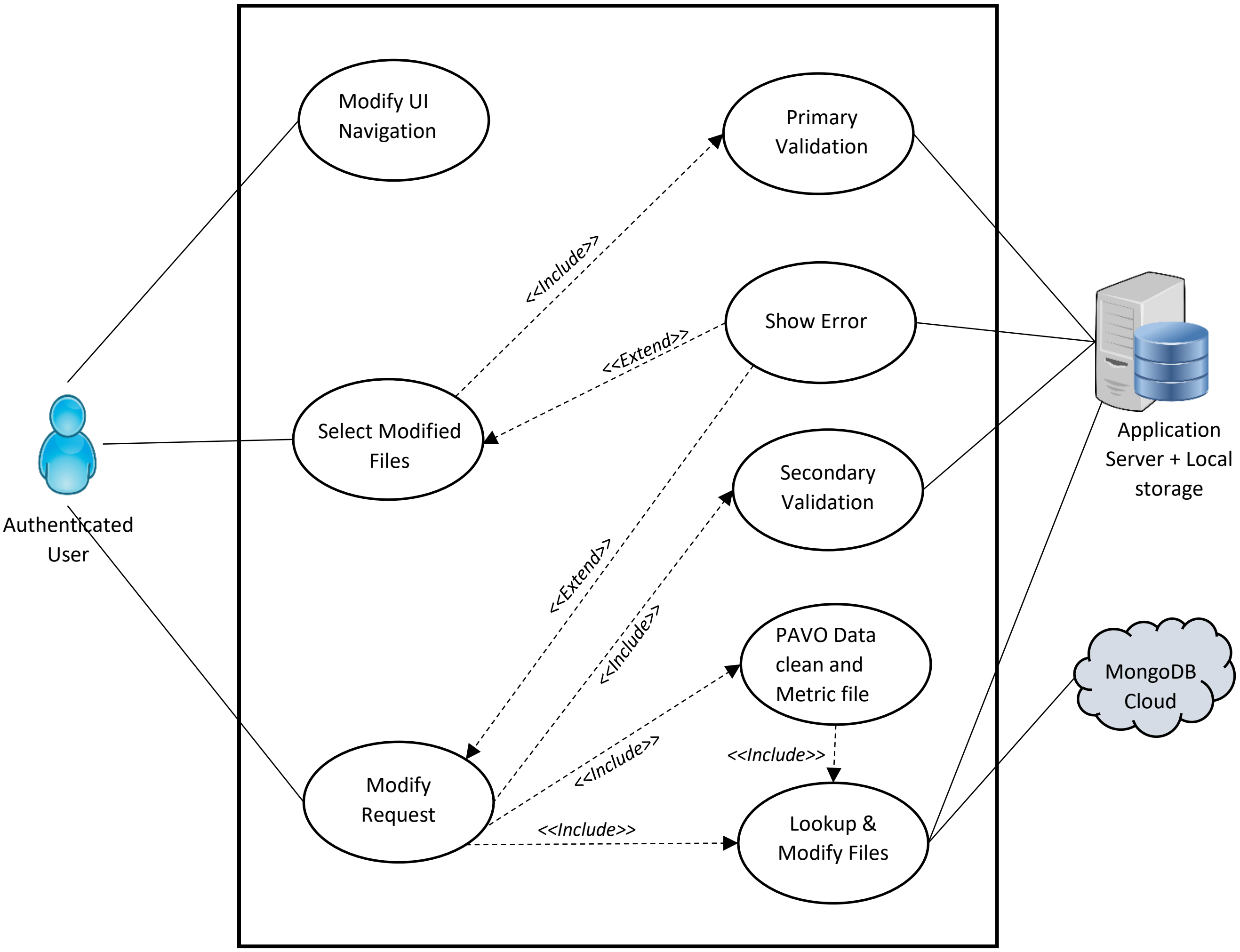

Fig 2.1.3: File modify Request Use case diagram

File Modify Request – Textual Description

Name:	File Modify Request
Participating Actor:	Researchers
Entry Condition:	Researcher want's modify data file from earlier submission.
Flow of Events:	1. Researcher request for data modification. 2. System displays earlier submission list of that researcher. 3. Researcher selects specific submission that he wants to modify. 4. System asks for the modified files (New raw data+ new metadata). 5. Researcher selects the new modified files (Raw files and metadata file). 6. Researcher uploads the files. 7. System validates the package. 8. System lookup for the old raw files that researcher requested to modify. 9. System replaces the old raw files with new raw files and replaces old metadata information with new metadata information related to the new raw files. 10. System notifies researcher about successful modification. 11. System computes metrics for newly uploaded raw files. 12. System stores the calculated metrics in the repository and releases the data.
Exit Condition:	Requested data is modified and pushed to the database for search and download.
Alternate Flows:	1. System finds error during package validation.

	a) System notifies the Researcher about errors. b) Researcher acknowledges the error message. c) System reverts and ask for modified files. 2. System finds corrupted files or files with negative values (less than -2). a) Dump the raw file and relevant metadata and notify user (No modification will happen)

File Search - Diagram

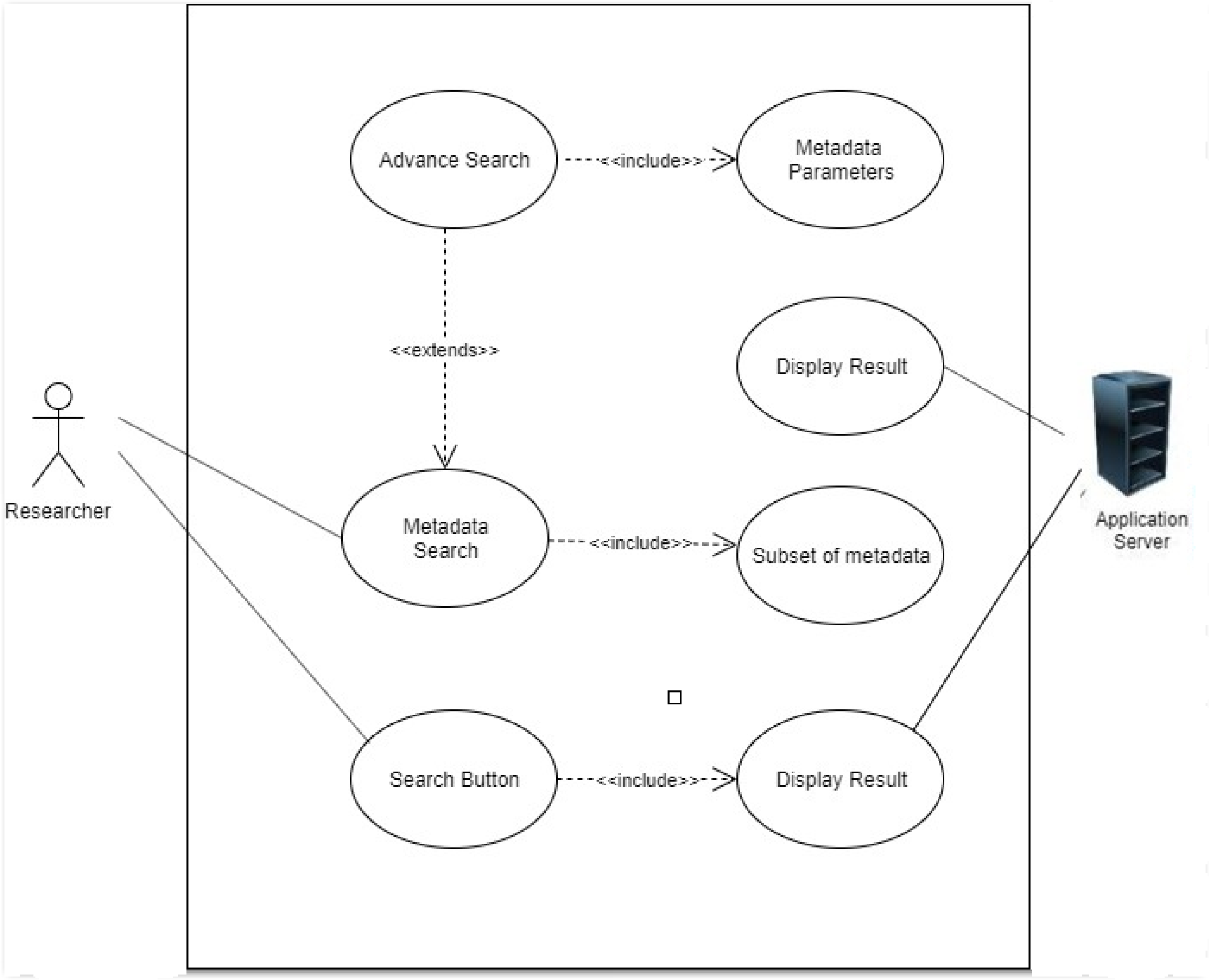

Fig 2.1.4: File search Use case diagram

File Search – Textual Description

Name:	File Search
Participating Actor:	Users (any users)
Entry Condition:	1. User is connected to the internet. 2. User is on Nature Palettes search page.
Flow of Events:	1. System shows the researchers a search page. 2. Researcher enters the search term and click on `Search` button. 3. System perform the search 4. System return the search result. 5. Researchers see the search result 6. Researchers can use the advance search option to narrow the search result. 7. System return the user a refined result of the raw data.
Exit Condition:	Search terms matches with the metadata are displayed to the researcher.
Alternate Flows:	1. Search terms entered by the researchers did not match with the metadata. a. System shows an error message. b. User reads and acknowledge the message. c. System return the user to the first step. d. Researchers enter another search term and click on `Search` button. e. System perform the search with new search term and return the result.

File Download – Diagram

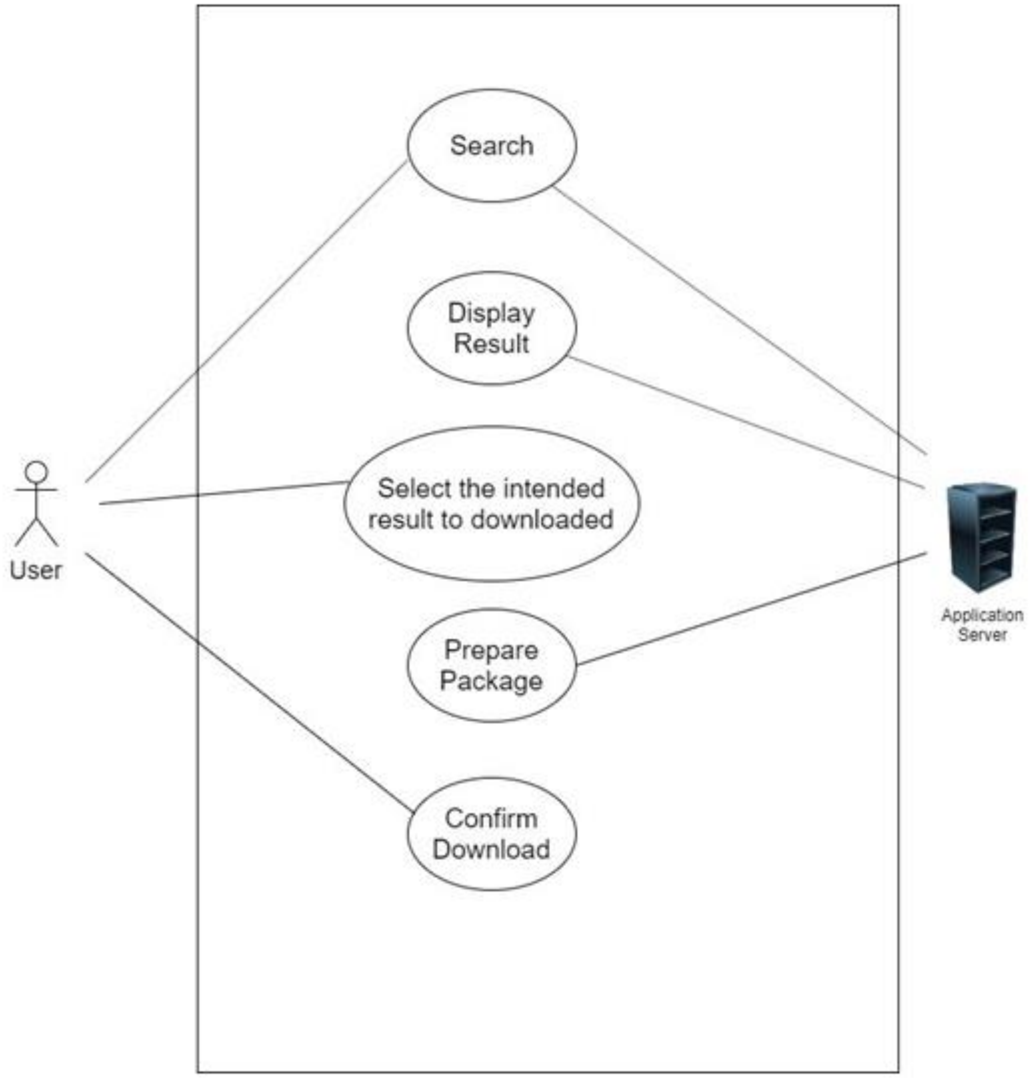

Fig 2.1.5: File download Use case diagram

File Download – Textual Description

Name:	File Download
Participating Actor:	Users (any users)
Entry Condition:	User performs a successful search query
Flow of Events:	1. System shows the results of raw data for the matched keywords. 2. User view the result and click on the Download button. 3. System retrieve the relevant raw files. 4. System generates a metadata file with the meta data records of all relevant raw files. 5. System will zip all the files (Raw files + meta data files) 6. System start downloading the zip file. 7. User will see a message saying that the file has been downloaded successfully.
Exit Condition:	A zipped file will be downloaded containing all the raw file and one metadata file (CSV) with all metadata records.
Alternate Flows:	6. Download is interrupted. a. System shows an error message. b. System will ask the user if he wants to re-try the downloading. c. Researchers clicks on Retry button. d. System started downloading the file again.

2.2. Design Goals

The system goal is to provide a feasible and more convenient solution for researchers to share their spectral data in online repository and to allow any interested users to search download it as they want.

User friendly interface/UI: In terms of user interface, it was designed to be user-friendly as the targeted users are not an IT experts. So, it will provide a convenient error free user interface to access system to upload the data files, search for data and to download, also to modify the uploaded data with some simple steps.

Currently, the system searches with Darwin codes, species name, country but the system has been designed in such a way that in future more parameters of searches can be added.

Access control: *Administrator, contributors*, and *beneficiaries* are the three main user types of the system. The administrator role wasn't considered for the first release of this software, as the main concern during the initial release was to demonstrate the main client requirements.

Administrators will have the full privileges through out the system, which will be implemented in future release.

Researchers will be allowed to upload the spectral data with meta data (If those two files match only).

Beneficiaries/other non authenticated users will have read-only level of privilege. They can search for any data and can download the raw data and meta data. They can't do any level of modification. In system perspective beneficiaries will be considered as guest users.

Flexibility and scalability: As a part of future enhancement, the search criteria can be exceeded. Moreover, as it is expected a huge set of data will be stored, the system has been designed to take over the load in the future. The storage scalability is there as the DB is deployed in cloud. Based on client budget can purchase more space.

Reliability: Nature's Palette system will be reliable in terms of persisting data on the permanent storage system (MongoDB) and be able to distinct different kinds of data. The searching algorithm will be consistent and optimized with the logics for effective response time.

Fault tolerance: In case of any error, failure or data validation errors the system will inform the user about the failure with a proper message/logs without affecting other parts of the working system.

Target environment: The program will be written in JavaScript to be able to run on Nodejs environment. The data will be stored in MongoDB.

Hardware and software mapping: In the production stage during the initial release the system deployed in two nodes (virtual). One node was used to deploy the web services also the raw files were stored locally in this node. The second node was used for DB. Mongo Db was deployed in the second node in cloud where the meta data records are stored. The basic idea was to run the web services and DB in two sperate nodes improve the performance in terms of resource utilization. The raw data and meta data are stored in physically isolated location so that one site impact wont harm the other node. In future the raw data backup will be stored in remote mongo DB and vice verse.

Security: As the Nature's Palette system is for academic/research purpose the client didn't concern much about the security. Such security aspects (Eg: DB security features, HTTPS access for web server) wasn't considered at this stage. But the essential security features are there. Contributors need to login by using the username and credentials given by *administrator*. *Beneficiary* user can access as gust no login required.

2.3. Logical Architecture

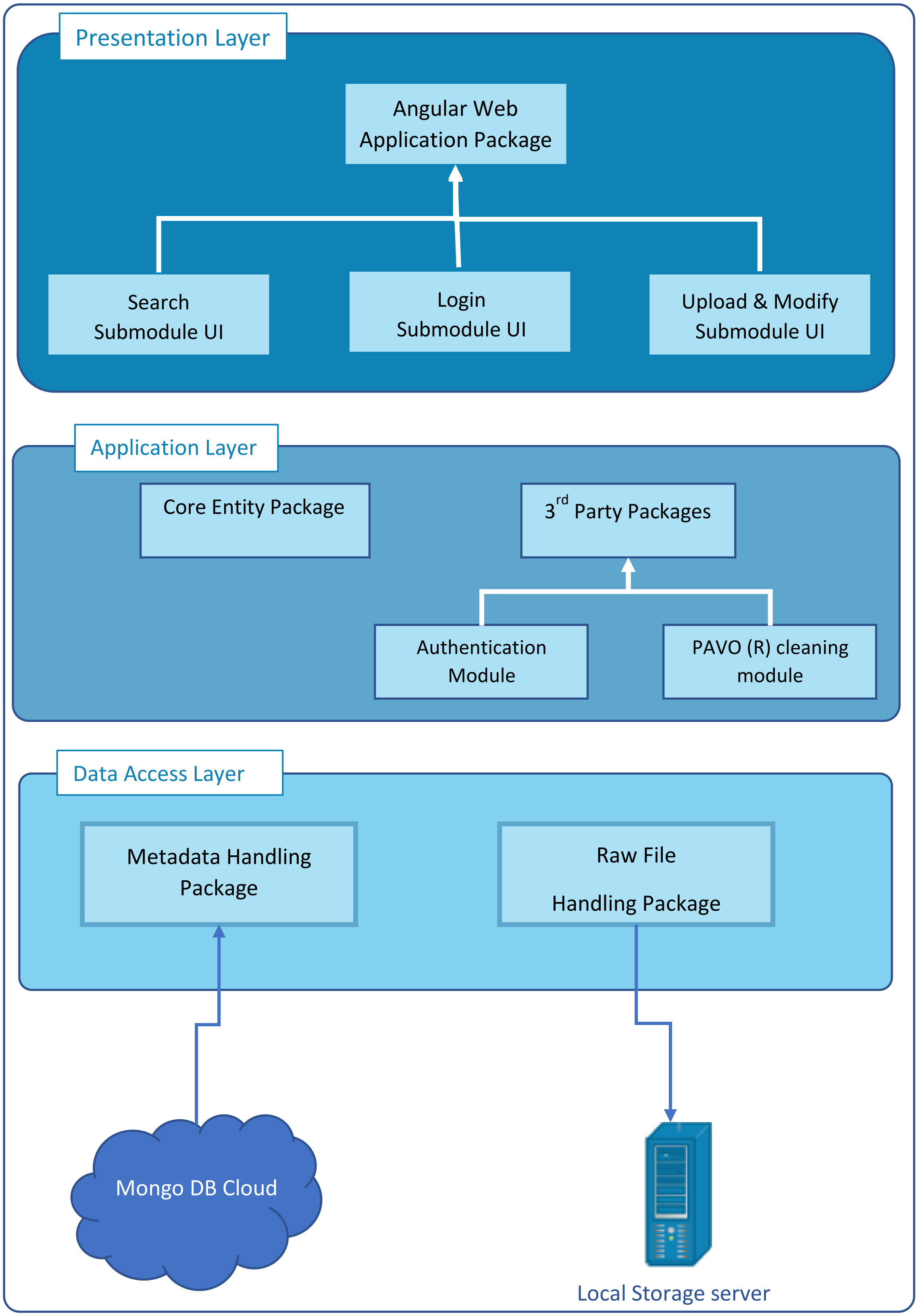

Fig 2.3: Logical Architecture

03. User Manual (Readme)

The Nature's Palette digital repository system was designed to be most user friendly for anyone to access it and get things done in few simple steps. Basically, it has four major functionalities that allows users to perform on data repository. For public users, user privileges are given as researchers and normal users. To login as a researcher, need an authentication and other users can simply access system as a guest.

In the repository the users can upload their research data (spectral data), can request for modification to modify earlier submission, can searcher for other researcher's data and can download the data based on search results. Only the authenticated researchers can upload data or request for modification from their earlier submission. Data search and download can be done by anyone, no need of any login or authentication. To be a user with researcher privilege, need to have an account locally or in ORCID (In testing phase).

In Nature's Palette digital repository home page, can see four icons in the middle of the page. First one is for data search, second icon to join in digital repository as a researcher (to create a new account and its still under development), third icon is to upload data file and fourth icon is to request for data modification. If user wants to return to home page anytime can press the home button at top.

Anyone can search for research data from search view. Once the search icon is pressed it will navigate to search view. In search view, can see thirteen key words to search for raw data. If a user wants to search for raw data user can type the search term in relevant key field (Eg: sex: male). If the search term matches with any of the data, those raw data files will appear in below view. User can review it and download all those files (raw data and meta data file) in zipped format.

To upload the data files, user (researcher) has to press the upload button which leads to upload view. Read the submission instruction and tic agree to start the submission. In submission step 01, provide the basic details (Which is mandatory) and make sure to mention the data type (reflectance/transmittance/irradiance) and source (field or museum) which is very important for primary validation. In submission step 2, select the meta data file and raw data file (should be zipped) and upload it for validation. If the files are error free, user will be displayed with validation successful message. Following that user can confirm the submission to submit those files for repository. If the validation fails users must check and upload the correct files. After the submission, the system will review the raw data files in background and check for any corrupted raw files or files with negative readings (Users don't have to wait for this). If any file found, those will be dumped (no data or reading is less than -2) and user will be notified.

In case if user (researcher) wants to modify the raw data or meta data from earlier submission, can request for modification from home page modify button. It will navigate to modify request view. Here user must select the from which submission they are going to modify and can upload the modified raw data files and meta data file. Those files will go through the two-step validation process (same as upload). On successful validation the system will look for old file to be modified and replace both raw data and metadata with new files and the notification will be sent to user. In case of validation failure, user has to check and re-upload those files.

04. Deployment Dependencies

The Nature's Palette digital repository system is an open source project developed for academic research purpose. The source code is available in GitHub and the link is provided in below section. Anyone who is interested can use the source code for further development. This section describes the deployment dependency to run the service from the source code.

It is a platform independent system where it can run on any OS from the given server (physical/virtual). But to compile the source code need to install several packages. The system developed based on MEAN stack approach (M:MongoDB, E:Express server, A:Angular, N:Node). MongoDB for data storage purpose which can be installed anywhere. Express for backend server, Angular for user interface and the Node provides the run time environment for JavaScript source code.

Based on that, the system was divided into two section as server side which do all the processing job and client side which provides the user interface to let users to interact with system and communicate that user requests to server.

Initially in the given server need to install the node to ensure the run time environment and npm to install necessary node packages (below commands are given for UBUNTU OS).

```
sudo apt install nodejs

sudo apt install npm
```

From the given git link download the source code.

To run sever side need to install the npm installation, for that navigate to the "NP-Express-Server" directory from the downloaded source code and run below command,

```
npm install

node server.js
```

To run client-side services, need to install angular globally,

```
npm install -g angular-cli
```

Then navigate to the directory "NP-Frontends/NaturesPaletteUI" from the source code files and run below command to run the UI services which will launch the website.

```
npm install

ng serve -o --host 0.0.0.0
```

By default, it will serve through 3335 port but it can be changed from the source code. The website can be accessed by using the server IP or it can be mapped to domain name and call it by URL.

05. Web Link

Git Hub Link:

https://github.com/SouravBOrion/Natures-Palette

Website link:

http://sc-5.cs.mun.ca/