

## **Project Phase 2: Elaboration Iteration 1**

Names: Mustafa Khan, Ravi Bhagat, Van Minh Ngai

Student ID: 102938384, 105140023, 104831535

Group Number: 7

Due Date: July 16th 2020

## **Phase 2 Plan:**

Our plan in the elaboration stage for the first iteration involves designing UML diagrams representing the code repository that will hold the backend java API for the website, looking into the CKAN java repository that will be used for dependencies which will be used to make the open-data API for Windsor. Furthermore, we plan on designing the web pages themselves that will serve as the portal for visitors. Other objectives that are to be done during this phase include adding the datasets to the website, doing unit testing to ensure the code is working as intended.

These objectives seem reasonable within the given timeframe of the project and the fact that it's the initial development cycle. Doing anything more at this stage will become too much to handle and unattainable, resulting in a lot of time loss and will not serve to benefit the City of Windsor and their taxpayers who are funding the project.

For the UML diagrams, we will design them so that they represent the complete finished and stable version of the product after many iterations have been passed. This is because we do not have the completed code required for the project, and thus will have to make certain assumptions, and going forward make adjustments to the code or diagram as we see fit.

The java code repository and the CKAN repository are something that is very difficult to accomplish within one iteration cycle since it requires a lot of dependencies, prototypes, and other variables as such it will not be fully completed within this cycle. It will depend entirely on the velocity of our team members and how much we get done along with the direction we take.

Adding the data sets to the website is an objective that is not too difficult, we do however want to make the data more appealing which means adding data visualization tools that will help stakeholders in making better decisions. This step will lead us into the second iteration of the project as the code necessary to connect the live data to the backend will take some time to develop.

The last two objectives for this iteration cycle, involve unit testing, and further developing the website pages. Both these two will be done simultaneously while we code. We will test the code using unit testing to ensure its working as intended and add web pages that are necessary as we go along. Adding these pages shouldn't be too difficult as there are many resources available to develop the pages in HTML/CSS etc.

## **Code Repository:**

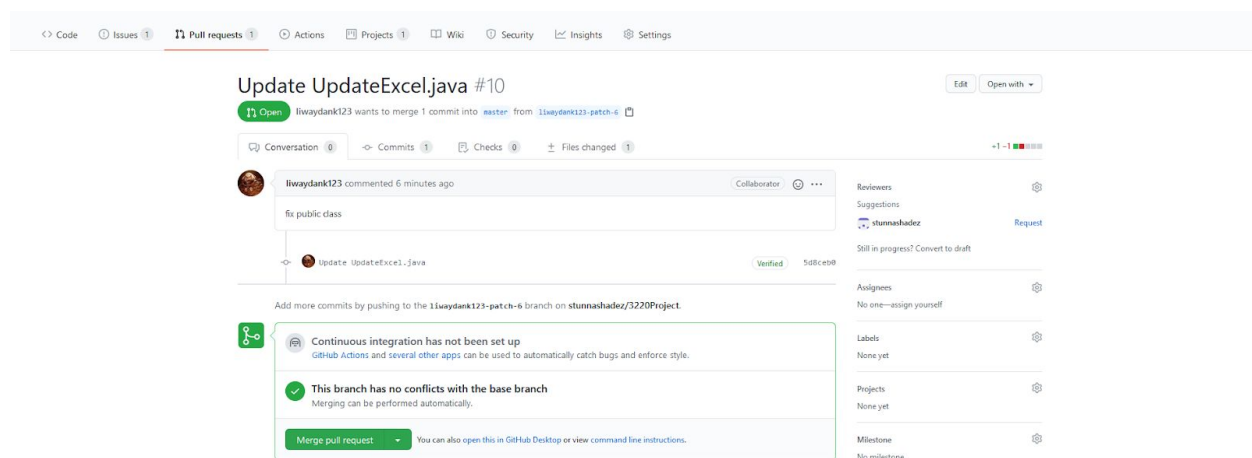
The code repository which holds all our java, HTML, CSS, XML, and other code can be found on our GitHub, along with all other files such as our UML diagrams, excel sheets, XML files, read me files, etc. The repository for the Open-Data Windsor Portal is still not completed and will be completed during the later iterations, which will hold more java API for the backend of the site through CKAN and java generated API. The GitHub code repository is located [here](#).

The Java repository for the Windsor open-data API will also be completed within the next iteration cycle as it has to encompass the CKAN and java generated API repositories (which are not completed within this iteration cycle). Once completed the Windsor open-data tool will allow developers to use the data and manipulate it any way they wish.

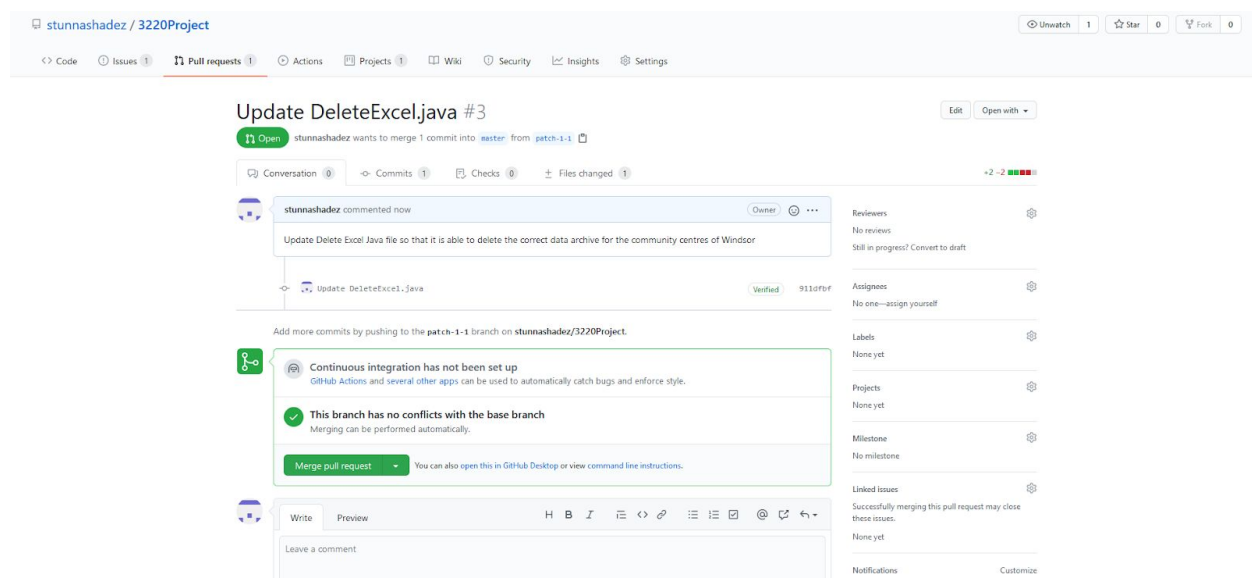
## Version Control:

The version control methods we employed are done through the use of GitHub's pull request feature. We had numerous versions and additions made towards the java code and other documents. Using pull requests we were able to allow different members of the team to make edits and then merge the edits into a completely new version that was satisfactory. We had collaborators create separate branches for code they wanted to put in and then merge the code into a newer version.

An example of a pull request and update of a version being performed is shown in the figure below in which code has been drafted by a collaborator, who then requests for a merger into the master branch, which is done after review from the other collaborators.



**Figure 1:** Pull Request being requested by collaborator to update version

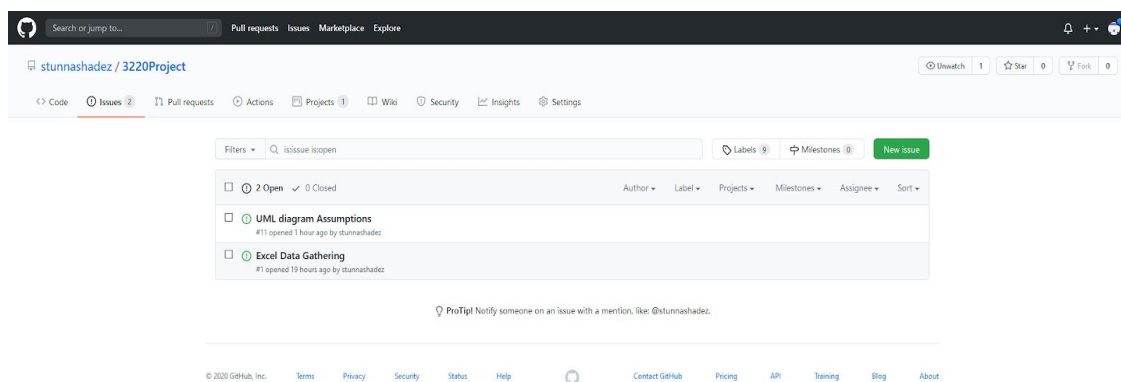


**Figure 2:** Pull Request being addressed and merged so that a new version can be created

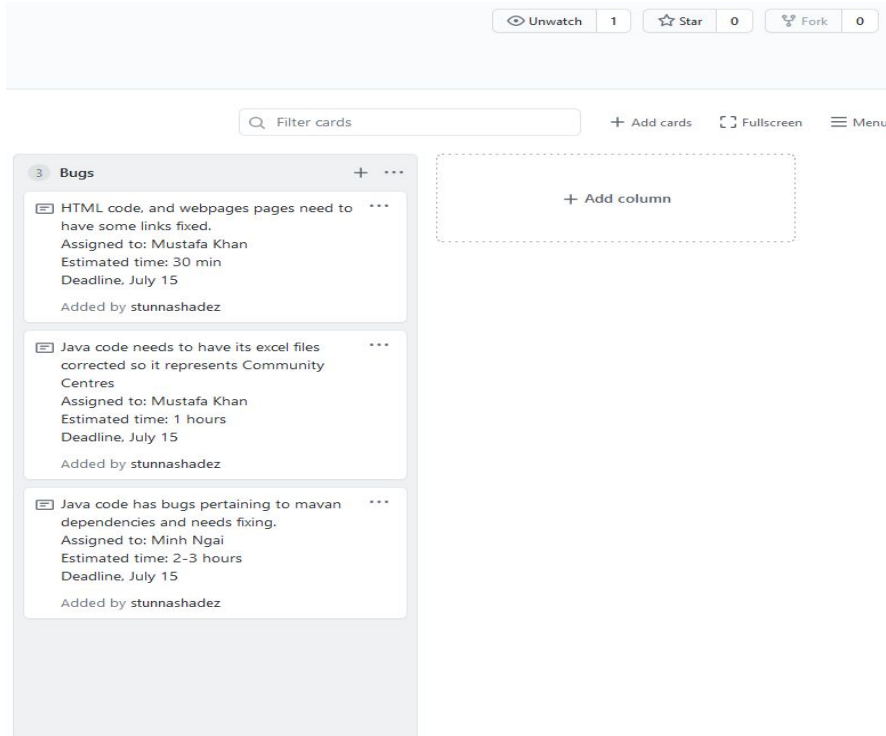
## Bug Reporting Tool:

For bug reporting, our team addressed this by using Github's Issues and Project Tools. For smaller non-coding related issues we used GitHub's Issues in which we dedicated to addressing strategic planning and directional issues that arose during the development of the project. For Coding related bugs we used the Project Tool. Within this tool, we created a bug report column where collaborators addressed bugs and how to handle them.

Specifically, our team encountered numerous bugs over the course of the timeline of this project. Most of the coding bugs we encountered came from the java code and some from the HTML code. Since the code and project are still in its first iteration phase, there are still some bugs to be fixed. Aside from coding bugs, there were directional issues which our team addressed as they came up throughout the cycle. Some of those directional issues are still being addressed and need to be looked further into when we begin phase 3. The major directional issue we face currently for the next iteration cycle includes how to implement the CKAN and java API for the website.



**Figure 3:** Non-coding related issues being tracked using GitHub's Issues Tab

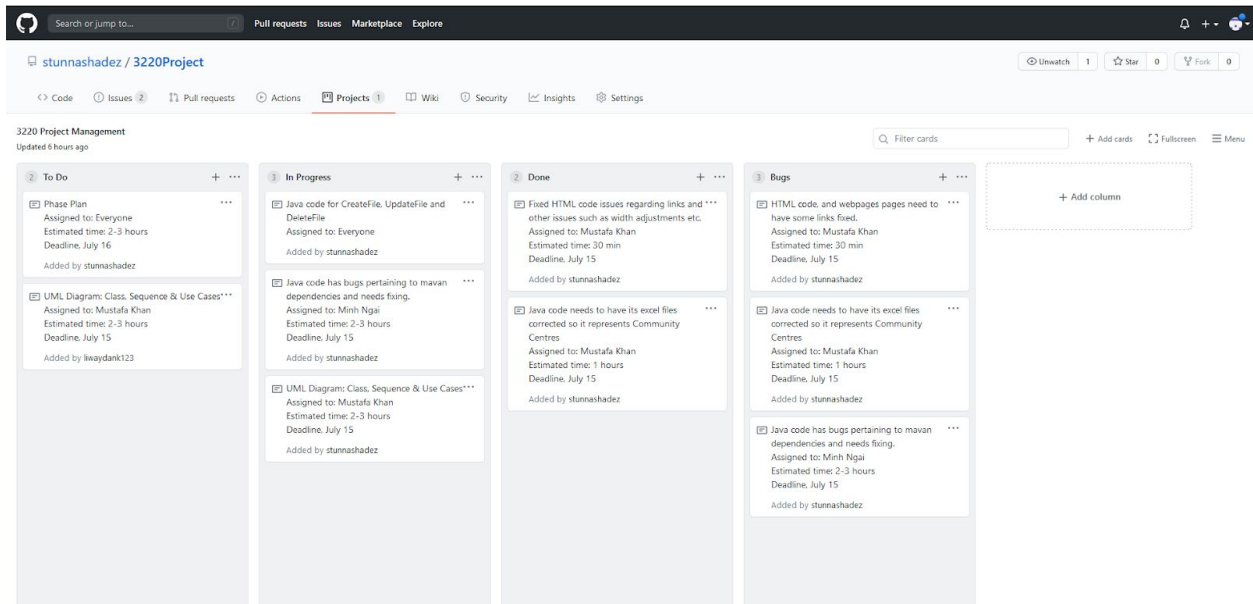


**Figure 4:** Coding related bug being tracked using GitHub’s Project Tab

### **Project Management/Task Tracking For The Team:**

The project management and task tracking tools we employed used GitHub’s Project feature. We created a “To-Do” column to track tasks that needed to be completed, a “In Progress” column to track which tasks were being worked on and by whom, and we created a “Done” column in which we tracked which tasks had been finished.

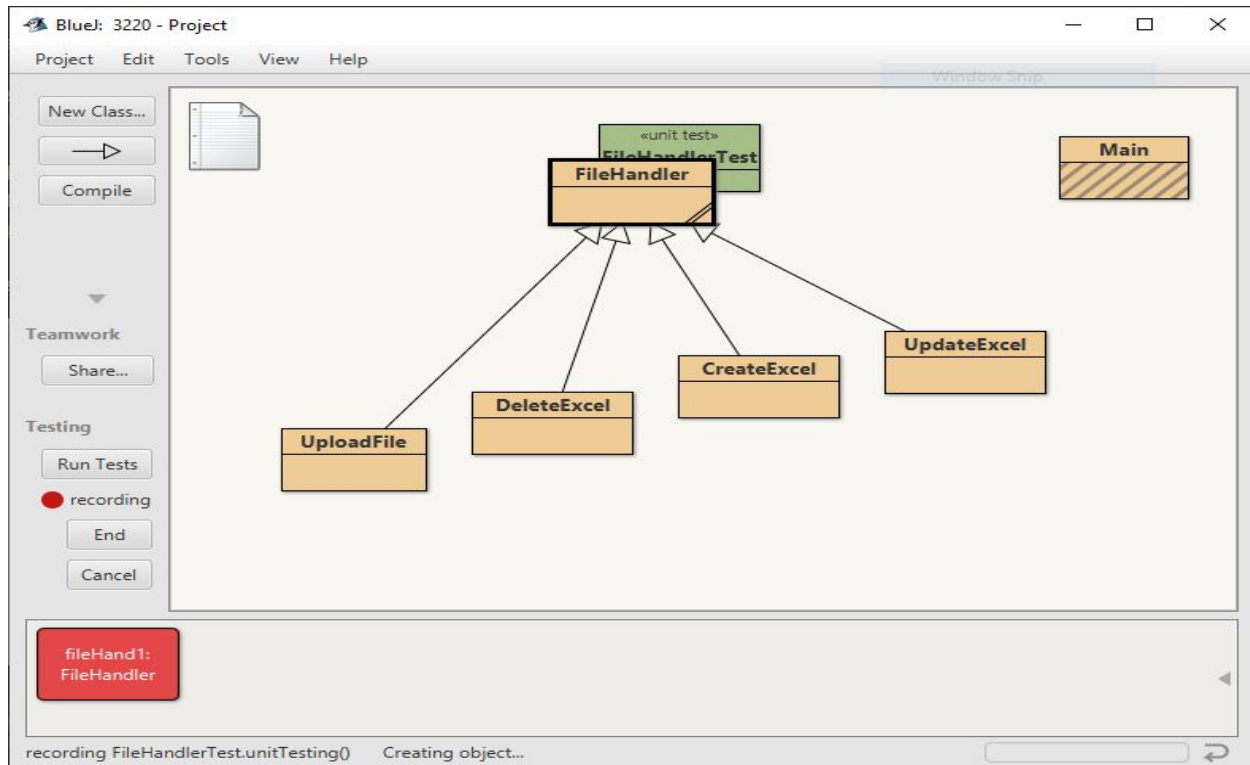
During the first iteration phase of this project, our task tracking and management were lacking a bit and we had trouble meeting deadlines and reaching targeted goals. However, during the end of the cycle, we were able to track relatively well and were able to set attainable goals which we would reach.



**Figure 5:** Project Management/Task Tracking managed Project Tools for collaborators in GitHub

## Testing Strategy:

For our testing strategy, we are using BlueJ to implement the Unit Testing for our project, and it should make our testing process more flexible in the future. As we can see in the screenshot below, the Unit Testing process of FileHandler allows us to make sure that our code works, even after adding merging the new code with the old or even after adding new code to the old code. If there is any problem with the code, BlueJ is powerful enough to let us know that there is something wrong with the code. One of the top qualities of it is that it allows visualization of what our class looks like in relation to other classes in the code. Important to point here is that BlueJ has only one relation that we can show. This testing will be done not only on FileHandler class, and it will also be done on other classes as well so that we can check all the classes and its variables. Therefore, we cannot add/see the dotted lines between Main and other classes when we call them from Main (look at the Class Diagram attached in UML diagram section).



## Documentation Repository:

For our documentation repository, we use README.md as a documentation source. In this file, we store the documentation about the java files, html codes, and information about the project. Also, we explain briefly about the java files functionality, and the project goal.

README.md

## 3220 Project - Open Source Data Website Documentation:

This project entails a website for the [City of Windsor](#) which holds the data sets for various industries allowing the stakeholders of the City namely the taxpayers to use the open data for making strategic decisions.

There are various documents within this project these include the:

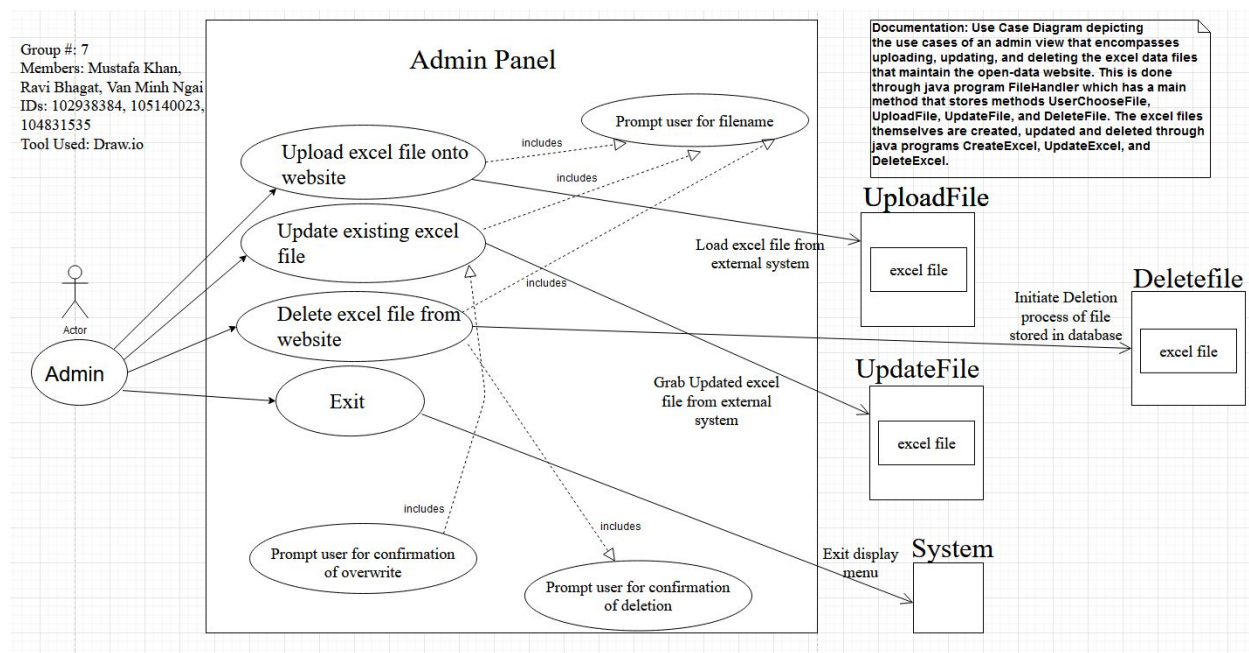
- Java files `CreateExcel.java`, `DeleteExcel.java`, `UpdateExcel.java`, and `UploadFile.java` which are capable of opening, adding, and deleting the excel sheets which maintain the open data for the website.
- The html files which display a sample of the website's pages.
- The initial pdf for phase 1 of the project, the inception report, which outlines the objectives for the project.
- An example excel sheet which represents the data block of a certain industry.
- The UML diagrams which represent the use case, class, and sequence diagrams.
- The `pom.xml` file which is used for the java file as a dependency for accessing excel files.

**Figure 6:** README.md file showing the documentations, instructions and information about the project

## UML Diagrams:

### Use-Case Diagram:

The use-case diagram below depicts all the use cases that are involved with the admin of the website who has the responsibility to upkeep the website and maintain the open-data catalog. The first case involves the admin uploading the excel document to the website (The excel document which has been created by the CreateExcel java file). The second case involves the admin uploading an updated version of the excel document (The excel document which has been updated by the UpdateExcel java file). The third case involves the admin calling a method to delete the excel document (The excel document which is to be deleted once DeleteExcel java file is called using UpdateFile method). The last case involves the admin exiting the java program. All of the three methods UploadFile, UpdateFile and DeleteFile are to be found in a main method within a java file called FileHandler. The use-cases essentially depict the admin modifying the data files which stores all the open-data in excel format. Since this data has to be constantly uploaded, updated and deleted, the admin has java programs to maintain the management of the data.

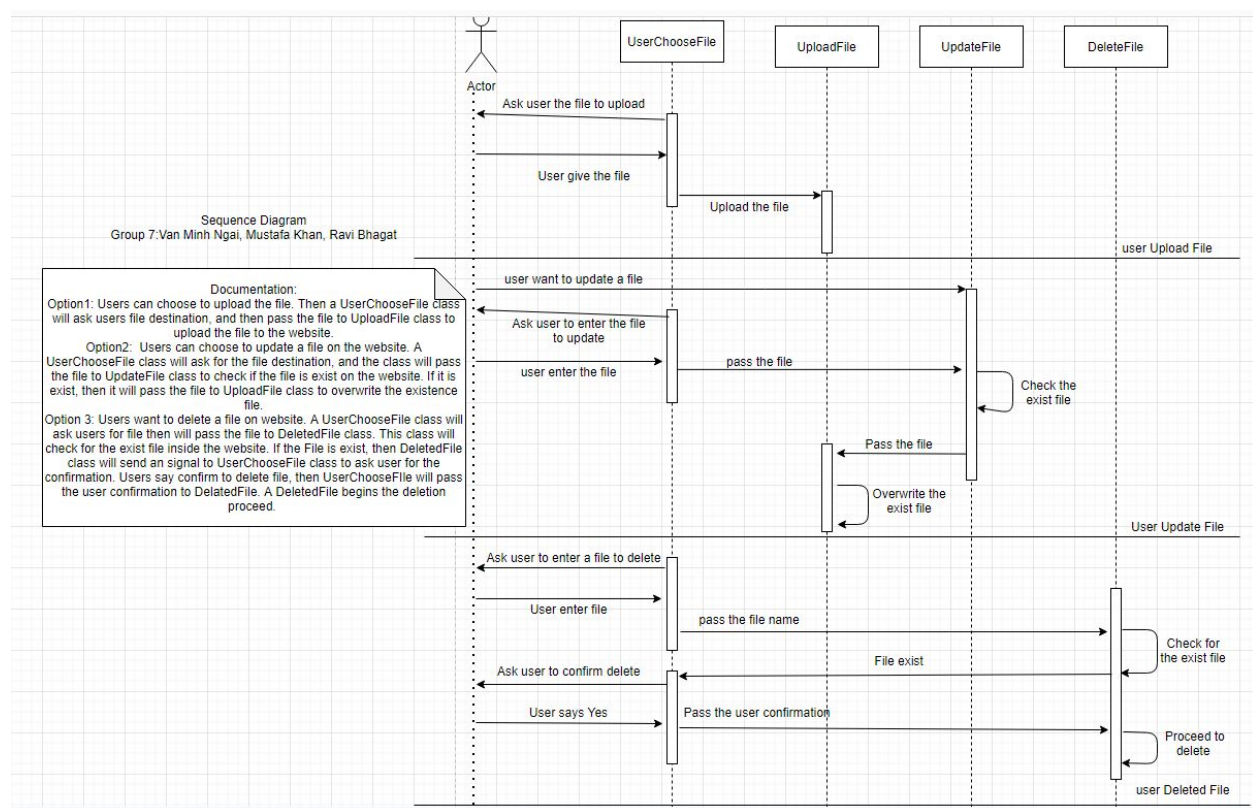


**Figure 7:** UML diagram depicting the use cases for admin view, done using draw.io tool

### Sequence Diagram:



The sequence diagram depicts the admin going through the sequence for adding, modifying, and deleting data files onto the website. The admin starts by being asked by the UserChooseFile to upload a file, after which the admin selects the file, the UserChooseFile accesses the UploadFile method to upload the file. In the second sequence, the admin is asked to update a file from which the admin enters the file name to be updated, which is then passed to the UpdateFile method that checks if the file exists, then it goes to the UploadFile which then overwrites the current file thus updating it. In the last sequence scenario, the admin is asked to delete a file, the admin responds with the file name, which is passed to the DeleteFile Method which checks if the file exists and then asks the admin to confirm the deletion, after confirmation from admin is received the DeleteFile method deletes the file.

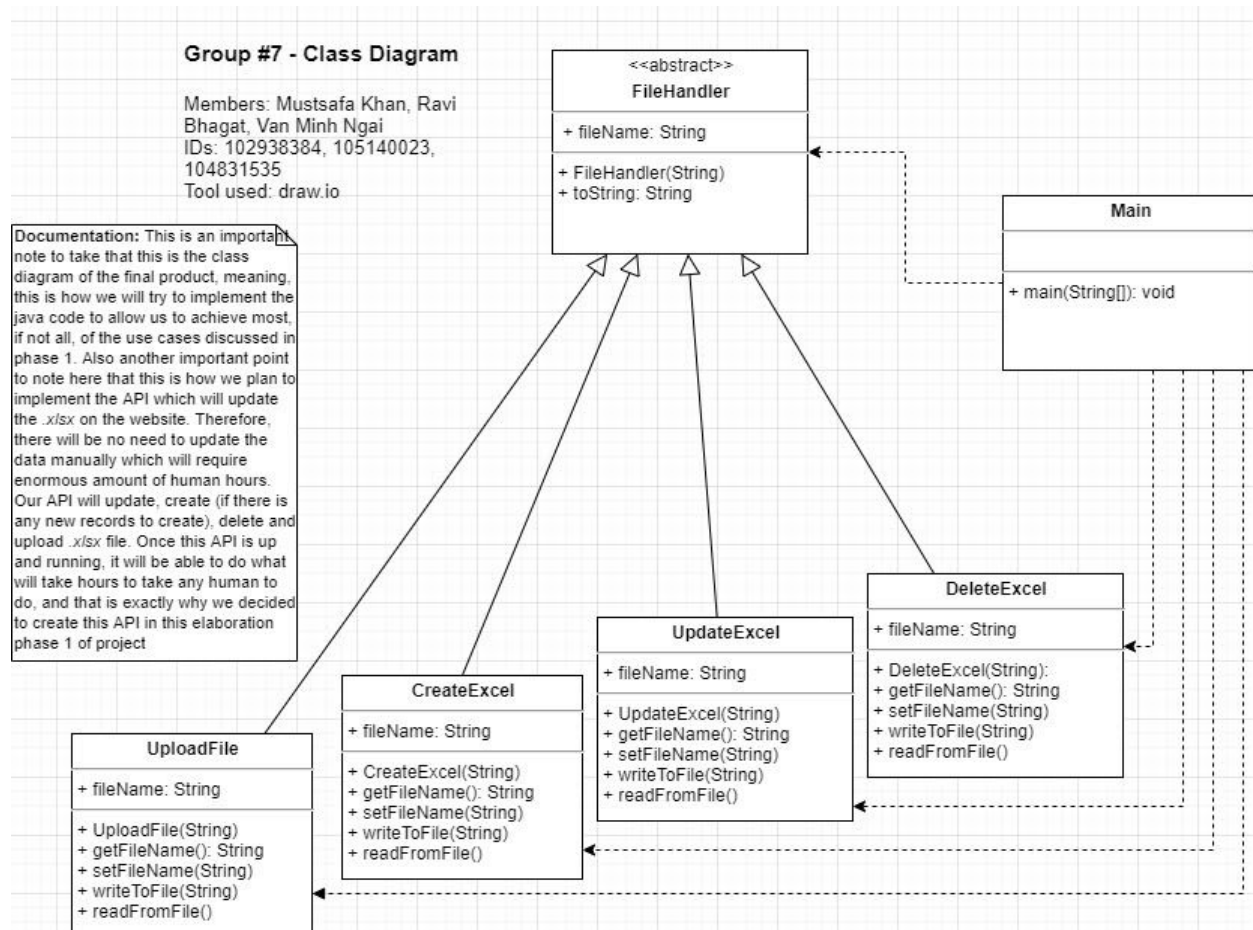


**Figure 8:** UML diagram depicting the sequence diagram for admin view, done using draw.io tool

## Class Diagram:

The class diagram below depicts the completed version of the classes that are to be used to handle the data files. The main class holds the classes for uploading, creating, updating, and deleting the excel data files. The FileHandler class acts as a superclass for uploading, creating, updating, and deleting excel files. Note that the class diagram represents classes and methods

that are to result from the final completion of the project and currently are not implemented exactly in the corresponding code as the code is incomplete due to it being part of the first iteration cycle as of yet.



**Figure 9:** UML diagram depicting the class diagram for admin view, done using draw.io tool

## Resulting Product:

Before we go ahead and discuss the resulting product, we have to mention due to the learning curve that comes with working with new tools, we were not able to finish the implementation of CKAN java dependencies, and other backend java API. This has only been our first iteration of the elaboration part of the project and what we have made so far is nothing short of progress. As we know, by implementing CKAN it gives us the ability to make an open data portal for us. Once we have these APIs up and running we should be able to add more categories to the website making the open data tool more complete.

With that out of the way, let us talk about the resulting product we came up with in the first iteration of this elaboration phase. We were able to set up the java backend which is able to create, update, delete, and upload excel files to the website. This allows the admin of the open-data web portal to be able to manage the data efficiently.

With that said we did have a lot of hurdles along the way, as our team was very inexperienced with working with CKAN and java API, we realized that the way we approached this iteration cycle using our current code is not correct because it does not work with many different formats other than XLS and XLSX. Thus for the next iteration cycle, we will need to adjust the code so that the CKAN and java library's own open data API can be implemented into it. Not only that but our team also realized later that our expected time to complete a task was much longer than the actual time needed and as such we ended up putting in more hours on objectives that we intended not to.

In terms of the website prototype itself, we were able to get some of the website's prototype pages done but not as much as we intended because as we stated earlier we ended up spending more time on other tasks than expected. Since we were not able to connect CKAN dependency and other java API we were not able to add data visualization to our website. The link to the website can be found [here](#).

All of these issues will be addressed in the next iteration phase in which we will implement the CKAN plugin and the other java open data libraries so that the website is able to host data sets in different formats and also have different visualizations for the data that is available.