**COMP-3220 GROUP PROJECT**

**ELABORATION PHASE 1**

Written By: Team 1

Matthew Muscedere

Hiba Rehman

Karanveer Sidhu

Matt Gagnon

Varun Teja Katakam

Blake Derksen

Tory Provenzano

**Iteration Phase Plan and Objectives/Deliverables**

**Objectives and Deliverables**

**Project Goal:** To develop an interface which allows users to view and track Covid-19 infection, death and vaccination data throughout Canadian provinces.

**Project Objective:** Designing a user-friendly UI which allows users to select regions, years, and data sets to view accurate information relating to the development of Covid-19 from early 2020 to late 2022.

**Deliverables:**

**User Interface:** The primary deliverable is the UI created using Eclipse JFrames to allow for user friendly interaction. The UI will also present the data fetched from the data repository on GitHub and display it to the user.

**Backend File Manager:** The secondary deliverable is the backend file manager of the UI (which is linked to the GitHub data repository), which correctly stores the data required for the UI to display the correct information to the user. The backend will also be responsible for proper error handling.

**UML Diagrams:** The UML diagrams created in this phase will primarily show how certain elements in our UI function, being more accurate than the ones created in the iteration phase previously.

**Phase Plan**

In this first elaboration phase, which is due on November 6th, we will primarily be composed of UML class and sequence diagrams that will describe the functionality of the project. This phase will also include the initial development of the UI and backend file manager, using JFrames to develop a very simple user interface that, while not looking aesthetically pleasing at this point, will include most of the essential elements required by the UI, creating a prototype interface.

In the second elaboration phase, which is due on November 20th, we will be shifting to a heavier focus on implementing the UML diagrams into the code repository on GitHub. Our effort will be put into further expanding our project, with 5 or so different use cases implemented, as well as adding/expanding on a visual user interface. We will also likely provide UML diagrams describing all these use cases and have everyone fully setup and implemented in our system of tools.

The final phase of the project is our final presentation and report, which is due on November 25th and November 27th, will demonstrate the functionality of our project, with explanations on certain aspects as well as screenshots as examples. The report itself will document how we designed, managed, approached, and developed the final project.

**UML & Case Documentations**

**1.) View COVID-19 Data by Region**

* **Description:** Allows users to select a specific region (province or territory) and view corresponding COVID-19 data, including infection rates, deaths, and vaccination statistics.
* **Actors:**
* **User:** Initiates the action.
* **Data Repository:** Stores COVID-19 data and provides it to the application.
* **Preconditions:**
* The UI is loaded, displaying options for region selection.
* The system is connected to the data repository.
* **Main Flow:**

1. **User** selects a region from the dropdown list in the UI.
2. **System** sends a request to the data repository to fetch COVID-19 data for the selected region.
3. **Data Repository** retrieves the relevant data and sends it back to the system.
4. **System** displays the retrieved data on the UI, showing updated infection, death, and vaccination stats for the region.

* **Postconditions:**
* The UI displays updated COVID-19 data for the selected region.
* Exceptions**:**
* If data retrieval fails (e.g., repository is offline), the system shows an error message.

**2.) Filter Data by Time Period**

* **Description:** Allows users to narrow down COVID-19 data by selecting a specific time range (e.g., months or years) to view historical or recent data trends.
* **Actors:**
* **User:** Specifies the time range.
* **Data Repository:** Provides filtered COVID-19 data.
* **Preconditions:**
* The UI is loaded with an interactive time filter.
* The connection to the data repository is active.
* **Main Flow:**

1. **User** selects a start and end date from the time filter on the UI.
2. **System** requests data for the specified time from the filtered data.
3. **Data Repository** processes the request and returns the filtered data.
4. **System** updates the UI to show COVID-19 data within the selected time.

* **Postconditions:**
* The UI displays COVID-19 data for the chosen time range.
* **Exceptions:**
* If data for the specified time range is unavailable, the system shows a “Data Unavailable” notification.

**3.) Error Handling and Notifications**

* **Description:** Manges error cases, such as network failures or data unavailability, by notifying the user of the issue.
* **Actors:**
* **User:** Interacts with the UI.
* **System:** Monitors and handles exceptions.
* **Preconditions:**
* User attempts to perform an action (e.g., selecting a region or filtering data).
* **Main Flow:**
* **System** detects an error (e.g., failed data retrieval).
* **System** displays an error message to the **User,** detailing the issue and suggesting possible actions (e.g., retry or check connection).
* **Postconditions:**
* User is informed of the error and possible solutions.
* **Exceptions:**
* If multiple errors occur, the system queues messages or displays a comprehensive error log.

**UML Diagrams:**

* [Class Diagram](https://lucid.app/lucidchart/03c5b142-7672-4cf5-aa26-cf2c38937654/edit?viewport_loc=-564%2C-1156%2C3103%2C2381%2C0_0&invitationId=inv_f47df850-e6d8-42c5-a901-73e1f36ab4e8)
* [Sequence Diagram](https://lucid.app/lucidchart/03c5b142-7672-4cf5-aa26-cf2c38937654/edit?viewport_loc=-1157%2C-566%2C2227%2C1709%2CGATlxTZdFI39&invitationId=inv_f47df850-e6d8-42c5-a901-73e1f36ab4e8)

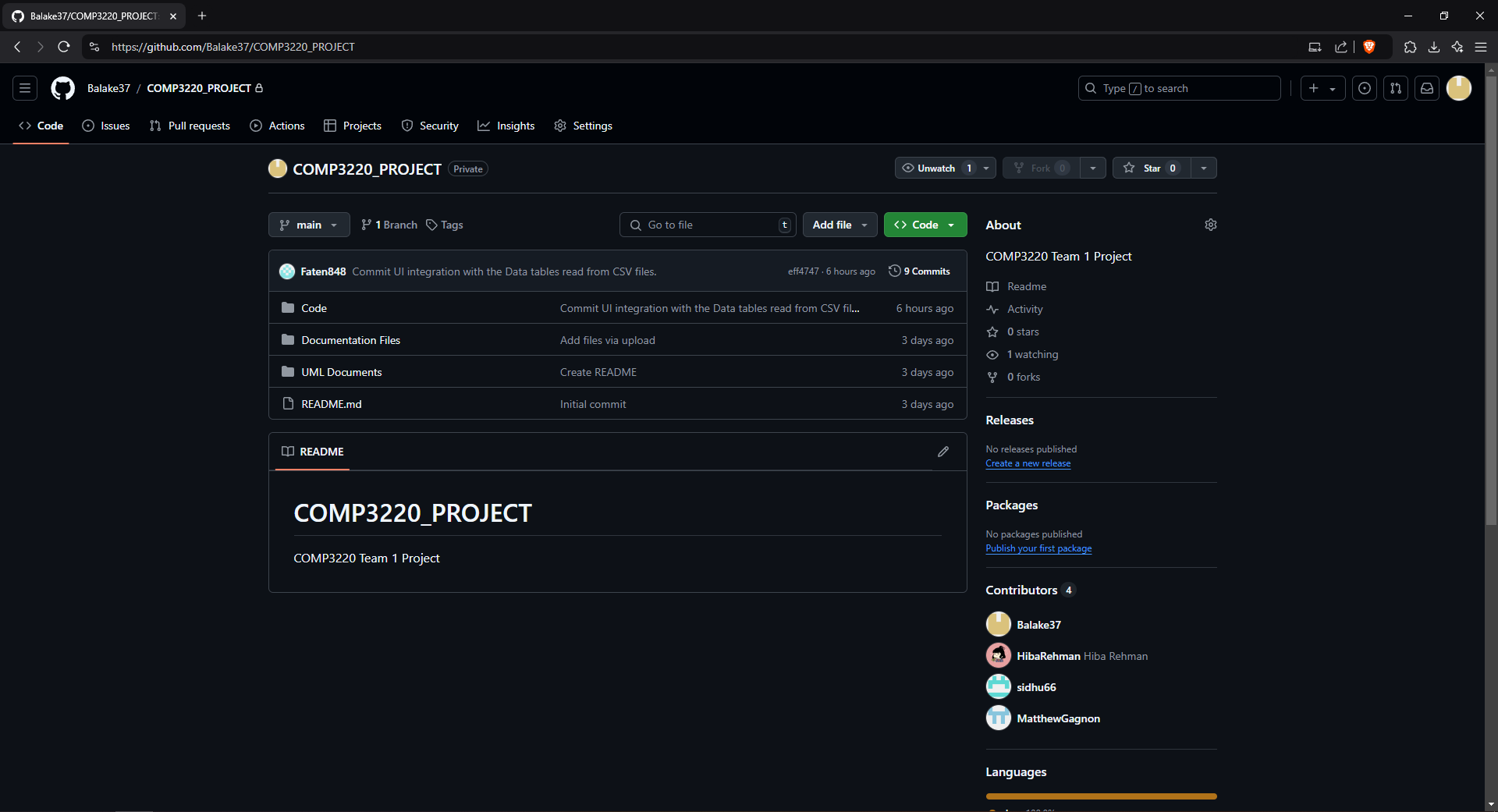
**Updated Diagrams:**

* [**updated class diagram**](https://lucid.app/lucidchart/42ec5ec5-2891-4ef2-9c78-eff0c1b6cb28/edit?viewport_loc=-865%2C-474%2C4378%2C1631%2C0_0&invitationId=inv_f6580e82-b17d-4b7e-a1e9-b88f018e18a0)
* [**Updated sequence diagram**](https://lucid.app/lucidchart/42ec5ec5-2891-4ef2-9c78-eff0c1b6cb28/edit?viewport_loc=-1010%2C-1087%2C3550%2C1323%2C4gVoh_KE7lu.&invitationId=inv_f6580e82-b17d-4b7e-a1e9-b88f018e18a0)
* [**Use case diagram**](https://lucid.app/lucidchart/42ec5ec5-2891-4ef2-9c78-eff0c1b6cb28/edit?viewport_loc=-1146%2C-297%2C3914%2C1458%2CdiVoDH_Xu.Gs&invitationId=inv_f6580e82-b17d-4b7e-a1e9-b88f018e18a0)

**Team Collaboration Tools and Resources**

**Tools**

The main tool that we are using to collaborate and easily work together is GitHub. We are using it as a repository for all our code, UML diagrams, and documentation. Our code is being made on applications like Microsoft Visual Studio Code, MyWeb, Overleaf, and JFrame. Here is a screenshot of our main page of our repository:



Here is a link to the GitHub in order to view our content we currently have: <https://github.com/Balake37/COMP3220_PROJECT>

The second most important tool that we are using is Office 365. The main tools that we are using from it are Microsoft Word and Microsoft Teams, but we will be using Microsoft PowerPoint later in the project. These tools are very important as they are our main catalysts for documentation and communication and allow us to easily work together by working in the same documents and messaging each other in our Teams group.

The tools we are using for making UML diagrams and documents are Visual Paradigm and Lucid Chart. These tools are very important in allowing us to easily create our UML diagrams, which are a cornerstone of development and helping our team understand our vision. Screenshots and links to our UML will be provided in the “UML & Case Documentations” section.

The last tool that we are currently using to our disposal is Redmine. It is something we are currently setting up with everybody, but it is a helpful tool in which we can report bugs and problems with parts of our project, as well as better manage our team roles/contributions.

**Resources**

The main resources from which we gather the data of COVID-19 Cases in Canada and show it off in our application are:

* <https://github.com/ccodwg/CovidTimelineCanada/tree/main>
* <https://github.com/ishaberry/Covid19Canada>
* <https://art-bd.shinyapps.io/covid19canada/>

**Next Iteration Plan**

**Objective for Next Iteration:**

* Strengthen the project by implementing additional features in the user interface (UI) and backend, expanding functionality based on the use cases and UML diagrams developed in the current phase.
* Integrate code and test the UI elements in a real-time environment to ensure stability and accuracy in data display.

**Key Goals:**

**1.) Implement New Use Cases:**

* Add 3-5 new use cases to cover more user interactions and data functionalities.
* Ensure each use case has corresponding UML diagrams for easy reference and implementation.

**2.) Expand User Interface Functionality:**

* Enhance the UI developed in this phase with additional elements to make it more user-friendly and visually engaging.
* Begin incorporating error-handling mechanisms and notifications in the UI to improve user experience.

**3.) Backend Enhancements:**

* Refine the backend file manager to streamline data handling and retrieval processes.
* Improve data accuracy and efficiency in loading information from the repository to the UI.

**4.) Testing and Bug Fixing:**

* Conduct usability and functionality tests on the UI and backend to identify and resolve bugs.
* Begin documentation for a testing strategy, including test cases for each use case to ensure all functions work as expected.

**5.) Documentation and Team Coordination:**

* Update all UML diagrams, case documentation, and user manuals to reflect any added features or modifications.
* Coordinate with the team using GitHub, Redmine, and Microsoft Teams to track progress, assign tasks, and review code changes.

**Deliverables for Next Iteration:**

* Complete set of enhanced UML diagrams for all new use cases.
* A refined version of the UI with additional interactive features.
* A stable backend manager that accurately processes and displays data.
* A detailed testing plan with documented test cases and screenshots.

**Contribution Table**

|  |  |
| --- | --- |
| **Part Contributed** | **Collaborator** |
| Code UI | Hiba Rehman |
| Code “Backend” File Manager | Matthew Muscedere, Karanveer Sidhu, Matthew Gagnon |
| UML & Case Documentations | Varunteja katakam |
| Team collaboration tools/resources | Blake Derksen |
| Next Iteration Plan | Varunteja Katakam |
| Iteration Phase Plan/Objectives and Deliverables | Tory Provenzano |