

Celyna Su (@celyna)
Jesse Garcia (@jgarc269)
Cristina Lawson (@cristina95138)
Enrique Munoz (@codeMasterHacker)
Luccas Chang (@LukeTheChang)

Lab 6 - It's Corona Time

[Data Set \(for bookmarking\)](#)

Features (analytics to implement) to Implement in Next Sprint:

- **World Data: (COVID-19.csv)**
 - **Feature 1:** as a user, I want to be able to see if the top 5 countries with the highest population age above 65 have the most deaths and cases to if older people are more susceptible to the virus
 - **Feature 2:** as a user, I want to be able to see the worldwide change from baseline average of various mobility analytics
 - **Feature 3:** as a user, I want to see if richer countries tested more that poorer countries to see if richer countries were able to handle the virus better than poorer countries
 - **Feature 4:** as a user, I want to see whether public transit mobility and workplace mobility have correlations with certain countries.

GUI design:

The GUI design for "COVID-19 Mobility Trends" features a light gray background with a central title. At the top, there are two buttons: "View cases vs. deaths analytics" (green) and "Import previous versions of data" (gray). Below these are six white panels with rounded corners, each containing a title, input fields, and a green "GO" button.

- Panel 1:** "View all countries by date:" with inputs for "Countries:" (dropdown), "From (YYYY-MM-DD)" (text), and "To (YYYY-MM-DD)" (text).
- Panel 2:** "View all confirmed cases and deaths for a country by date" with inputs for "Countries:" (dropdown), "From (YYYY-MM-DD)" (text), and "To (YYYY-MM-DD)" (text).
- Panel 3:** "View the total amount of tests and population for a country" with a "Countries:" (dropdown) input.
- Panel 4:** "View all countries with a health index less than the one chosen" with a "Health Index:" (dropdown) input.
- Panel 5:** "View the baseline fields of a country" with a "Countries:" (dropdown) input.
- Panel 6:** "View all countries with 65+ age population percentage greater than the one chosen" with a "Percentage:" (dropdown) input.



User test cases:

- **Feature 1 Test Cases:** as a user, I want to be able to see if the top 5 countries with the highest population age above 65 have the most deaths and cases
 - **Test Case 1:** Our application finds the top 5 countries with the highest population age above 65
 - Correct Output: We will have a 5 double bar graphs, where the x-axis labels will be the countries with highest population age above 65
 - **Test Case 2:** Our application finds displays the total number of deaths for that country thus far
 - Correct Output: We will have a 5 double bar graphs, where the the left bar represents the total number of deaths for that country thus far
 - **Test Case 3:** Our application finds displays the total number of cases for that country thus far
 - Correct Output: We will have a 5 double bar graphs, where the the right bar represents the total number of cases for that country thus far
- **Feature 2 Test Cases:** as a user, I want to be able to see the worldwide change from baseline average of various mobility analytics
 - **Test Case 1:** As a user, I want to visually see how the world has been affected by Coronavirus
 - Correct Output: A graph showing the worldwide average of mobilities
 - **Test Case 2:** As a user, I want to see changes to the mobilities of various outings
 - Correct Output: A graph showing the worldwide average of various mobilities
- **Feature 3 Test Cases:** as a user, I want to see if richer countries tested more that poorer countries
 - **Test Case 1:** We will have a two pie charts, where the right pie chart displays the top ten richest countries and their respective total number of tests

- Correct Output: A pie chart showing ten slices of the richest countries and their appropriate proportions.
 - **Test Case 2**: We will have a two pie charts, where the left pie chart displays the bottom ten poorest countries and their respective total number of tests
 - Correct Output: A pie chart showing ten slices of the poorest countries and their appropriate proportions.
- **Feature 4 Test Cases**: as a user, I want to see whether public transit mobility and workplace mobility have correlations with certain countries.
 - **Test Case 1**: As a user, I want to select which country to compare their transit and workplace mobilities.
 - Correct Output: A drop-down list shows the available countries to select from.
 - **Test Case 2**: As a user, I want to see the graph showing the mobility trends between the two desired fields.
 - Correct Output: A button titled “submit” is available for the user to press.
 - **Test Case 3**: As a user, I want to see the graph of the mobility trends between workplaces and public transits.
 - Correct output: After pressing the submit button, the app takes me to the webpage showing the desired line graph. A button is available for the user to return to the search operations webpage.

Taskboard:

Done list of last sprint:

- Worldwide cases vs deaths analytic
 - Implemented by Jesse Garcia
 - Updated SearchOperationPage.jsp
 - Created form that displays a message and a submit button to the user
 - Created connection to send the user to the new page using the submit button
 - Updated CovidFile.java
 - Implemented getCases function to get the total number of cases from all the countries in all months.
 - Implemented getDeaths function to get the total number of deaths from all the countries in all months.
 - Updated SearchOperations.java
 - Created connection between the front end to the server using the submit button
 - Submit button checks off a boolean value to calculate the analytic
 - Use getCases and getDeaths to calculate the analytic.
 - Calculate the percentage of people that die from the data
 - Send the user to the new page after calculation.

- Created caseVsDeathsPage.jsp
 - Created Connection to receive the data from the server to implement into the graph.
 - Implement jsp to display the graph to the user.
- Country cases vs death analytic
 - Implemented by Cristina Lawson
 - Created countryCasesVSDeathsPage.jsp
 - Implemented ChartJS chart for the frontend
 - Updated CovidFile.java
 - Implemented getCountryCases function to get the total cases from specific inputted countries
 - Implemented getCountryDeaths function to get the total deaths from specific inputted countries
 - Updated SearchOperations.java
 - Implemented frontend way for the user to choose the country and press submit
 - Communicates with backend to bring user to the analytics page and show the respective graph
- Change from baseline mobility analytic
 - Implemented by Jesse Garcia
 - Updated SearchOperationPage.jps
 - Created form that displays a message, drop down box and a submit button to the user
 - Created connection to send the user to the new page using the submit button
 - Created connection to send the data from the drop down box to the server after the submit button
 - Updated CovidFile.java
 - Implemented a getMobilityAvg function for grocery and Pharmacy, park, residential, retail, transit stations and workplace
 - Function takes in the data from the drop down month to filter the data by the user choice of month.
 - Updated SearchOperations.java
 - Created connection between the front end to the server using the submit button and to receive the data from the dropbox front end
 - Submit button checks off a boolean value to calculate the analytic
 - Implement function to send the data to the new page by mobility
 - Implemented a function to send the user to the new page after calculation.
 - Created allMobilityPage.jsp

- Receive the data from the server to implement into the graph.
- Added 6 different sections to the graph to display all the data at once.
- Implement jsp to display the graph to the user

To Do task list for the next sprint:

- Perform analytics to implement
 - Jesse
 - Implement specific mobility comparison to cases per month
 - Need to update searchOperationPage.jsp to allow the user to choose a month and a specific mobility
 - Need to implement a submit button to grab the data and send it to the server
 - Need to send the user to the next page
 - Need to update the searchOperationPage.java to retrieve the data sent from the front end.
 - Need to use the data to calculate analytic comparison
 - Need to send the data back to a jsp file
 - Need to Create new jsp file for new analytic
 - Need to retrieve the data sent from the server
 - Need to use the data to implement into the graph
 - Need to display the graph to the user.
 - Cristina
 - Implement worldwide change from baseline average of various mobility analytic
 - Create a new analytic frontend and backend
 - Update UI
 - Implement a new interface that is more in-line with our GUI designs
 - Update searchOperationsPage.jsp
 - Update displayResultsPage.jsp
 - Update editFilesPage.jsp
 - Update index.jsp
 - Lucas
 - Implement/complete analytic for comparing mobility trends between public transit and workplaces
 - Complete workPlacesVSTransportation.jsp file
 - Consider scrapping gson
 - Revamp functions in development in searchoperations.java due to possible data type issues

- Celyna
 - Design UI
 - Implementation
- Enrique
 - Implement Feature/Analytic 1
 - Add another function to CovidFile.java that performs this analytic
 - Add another .jsp file that displays the double bar graph chart
 - Update searchOperationsPage.jsp so that the correct response attributes are saved
 - Implement Feature/Analytic 2
 - Add another function to CovidFile.java that performs this analytic
 - Add another .jsp file that displays the the two pie charts
 - Update searchOperationsPage.jsp so that the correct response attributes are saved