Verse

Design Document

Team 9

Jackson Oriez
Nguyen Do
Lisa Le
Jacquie Yu

Index

Purpose

Application Description:

For some, privacy is a top concern, but for others, it's merely a peripheral concept bolstered by the "Nothing to Hide" argument. In fact, the vast majority blindly trust companies and click "I accept" out of carelessness or ignorance. However, in an age when people are more often online than not, we just want our user to be educated before they make a decision about their increasingly growing data. Verse is our attempt to help people get "versed" on how much and what kind of data companies such as Google, Facebook, and Apple take from them.

The purpose of this project is to create a web application called "Verse" which helps educate users on how their data is taken and used. The user will choose from companies such as Google, Apple, Facebook, etc. and upload their zip files obtained from said companies' data dumps to our website. Our site will then parse through the data that was input and give the user a visualization and a run through of all the data that was given to us.

Functional Requirements:

Website

<u>As a user,</u>

1. I would like to access the application through a web browser
2. I want to feel safe knowing that my data is encrypted with HTTPS (time permitting)
3. I would like to intuitively navigate through the app with a menu bar and links on the side
4. I would like to click on different tabs to analyze my Google, Facebook, or Apple data separately
5. I would also like to see a tab that combines all three data sources and their categories
6. I want to be able to click a downwards arrow or plus sign on any category to expand the details

User Profile (if time permitting)

<u>As a user,</u>

1. I would like to register for an account with my email

2. I would also like the option to access the service without registering an account
3. I want to be able to log in to my account
4. I would like the option to delete my account
5. I want to modify my account information, such as my password
6. I want to be able to come back and upload more data dumps to my existing data analysis report (given that I have an account)
7. I would like to know that my data is secure and will not be shared
8. I would like to understand why I should upload my privacy data to the app
9. I want to have access to a page that gives a summary of various types of Terms and Conditions Agreements
10. I would like to be able to clear my data upload history

Instructions for Obtaining User Data

<u>As a user,</u>

1. I would like to have specific instructions detailing how to use Verse
2. I want to be able to know where on the website to go to find the instructions
3. I want to be provided with step-by-step instructions including pictures on how to request and download my data from Google Takeout
4. I want to know which sections from Google Takeout can be downloaded or are recommended
5. I want to be provided with step-by-step instructions including pictures on how to request and download my data from Facebook
6. I want to be provided with step-by-step instructions including pictures on how to request and download my data from Apple

Data Dump Upload

<u>As a user,</u>

1. I would like to specify where I am uploading my data from (Google, Facebook, Apple)
2. I want to be able to upload multiple files from Google at once to Verse's online portal
3. I want to be able to upload multiple files from Facebook at once to Verse's online portal
4. I want to be able to upload multiple files from Apple at once to Verse's online portal
5. I want to be able to remove some files from the file upload portal

Data Parsing

As a user,

1. I would like my data to be automatically parsed by the website
2. I would like my to be able to submit my Google data for parsing
3. I would like my to be able to submit my Facebook data for parsing
4. I would like my to be able to submit my Apple data for parsing
5. I would like my data to be parsed and stored as many different categories
6. I would like to be able to receive email notifications that tell me when the parser is done analyzing my data (Extra, if time allows)

Visualization of Important Data

As a user,

1. I want to be educated about the various buzzwords and terms often used when describing privacy, including their definitions and example contexts
2. I want to see bar graphs and pie charts that compare the metrics of the data sources that I have uploaded (Google and Facebook if I have only uploaded those two, Google, Facebook, and Apple if I have uploaded all three, etc)
3. I would like to view a list of my data uploads on the analysis report to know what data the report is based on
4. I would like to clearly understand the significance and impact of my data
5. I would like to see a numerical snapshot of my data from Google including size, category, and date created
6. I want to be able to filter through my Google data via checkboxes to choose categories such as Photos, Maps, Gmail, etc
7. I would like to view my geographical data from Google Maps rendered on a map
8. I would like to see a numerical snapshot of my data from Facebook including size, category, and age of data
9. I want to be able to filter through my Facebook data via checkboxes to choose categories such as Photos, Friends, Messages, etc
10. I would like to view my friendship data on a graph that details length of friendship, number of interactions, totals of "likes", etc
11. I would like to see a numerical snapshot of my data from Apple including size, category, and age of data

12. I want to be able to filter through my Apple data via checkboxes to choose categories such as Music, Movies, Devices, etc
13. I would like to view my music data on a graph that details most frequently listened to songs and artists, etc
14. I would like to see a “time” vs. “amount of data” bar graph
15. I want to be able to export my summarized data analyses to my own machine

Non-Functional Requirements:

Architecture

As a developer,

1. I would like to use the React JS as a front-end
2. I would like to use Python and Django for the backend
3. I would like to use MySQL for the database

Usability

As a developer,

1. I would like the user to understand what Verse wants to achieve when first browsing the site
2. I would like the user to feel comfortable giving Verse their data
3. I would like to give the user a brief and easy to understand summary of their data after they upload it
4. I would like to give the user the option of saving their summary on their own device or get it sent to them through email
5. I would like my message that companies have a lot of data on people to be transparent to the user

Security

As a developer,

1. I would like the user to understand that their data will only be stored temporarily on our server when the user uploads it
2. I would like the user to understand that their data will be deleted from our database upon exiting the website or logging out.

3. I would like the user to rest assured that our services will not store any information from the user’s data dumps past the user’s session.

Hosting

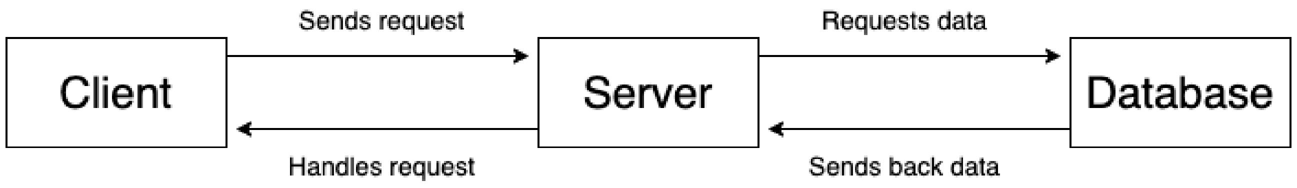
As a developer,

1. I would like to use Heroku to host the React app.
2. I would like to develop in python which will allow me to use Django as a web framework.
3. I would like to be able to store the user’s profile and summary data for them to access again at a later date (Extra, if time allows)

Design Outline

High Level Overview:

Verse is a web application that allows users to upload their personal data dumps from Google, Facebook, and Apple in the form of zip files to receive a summarized report of their privacy data as stored by these companies. These reports, which include graphs and other data visualizations, can then be downloaded for the user’s perusal along with their raw data that will be re-categorized to be consistent with the analyses.



Design Decisions:

This web application adheres to the multi client-server model. Data is transferred from the web page, which is built with React, via a POST request to our backend, which is built with Django. The server will accept the POST request, unzip the files, and input the data into the MySQL database where it will later be referenced by the parser. Once the parser has completely scanned through all the data, it will be packaged into logical groups which can then be fetched via GET requests by the front end user.

The database must have a different table for each source of information because Google, Facebook, and Apple will each format their data in different ways. For the sake of data privacy, the database will only store the files from each data dump, and once the parser has generated its final report, the data will be dropped.

Likewise, a different parser is required for each source as well so that the referenced file paths can be static. A cumulative parser will also be needed to compare data between the available data sources. These parsers consist of Python scripts that parse JSON file formats and assign their values to local variables, compute numerical totals for each category and variable of interest (size, time stamps, etc), and provide these totals in the form of an API that the front end can use to render the corresponding data visualizations.

Once the data has been categorized and packaged by the backend, it will go through a function in the frontend that shall be called the analyzer. The analyzer will compare numerical values and generate statements that explain the significance of the values as computed.

System Components:

- Web Client
    - The main interface for our application runs on a website accessible from any web browser.
    - The user will send POST requests to the API server to upload zip files of their data dumps.
    - After some processing time, the user will then be able to send GET requests to retrieve the data report and visualizations from the API server.
- API Server
    - The API server exposes all the information necessary for the front-end rendering of the data visualizations with React.
    - It will also handle HTTP requests from the user and pass data onto the Parser.
    - This server will never persistently store the raw data that is uploaded by the user, it only keeps them locally to be passed onto the Parser.
- Data Parser

- The Data Parser receives raw data from the API server and runs it through various programs that will extract files from the master zip file(s).
- It will also parse the JSON data into local variables, and sort the raw data into categories and filters such as time, type, and size.
- The Parser then passes this data to the Data Analyzer.

- Data Analyzer
    - The Data Analyzer takes the information given by the Parser and computes relations based on any given variable.
    - The Analyzer is also responsible for coming up with numerical comparisons between the various companies. For example, Apple stores “this much” less data than Facebook within “this” category.
    - The information provided by the Analyzer is used to populate the graphs and charts that compose the data report.
- Database
    - The Database holds any data that needs to be preserved for later use.
    - Data that may be held in the database includes past upload metadata, temporary data from the data dump, and user data.
    - The Database holds all the metadata provided by the user and is stored there for reference by the server until the parsing is done, after which it will be deleted.

Design Issues

Functional Issues:

User Accounts and Authentication

Should Verse even offer the option to create an account?

This would mean keeping more data and it might not be good to become THE company that ends up holding a master data set.

Potential Solutions:

1) **Skip creating accounts, allow users to export their data report, and have them be responsible for keeping it themselves**
2) Use existing login options like through Google, Facebook to create a new account
3) **Create our own login feature using Heroku/Django that includes username, password, profile, and history of uploads**

Decision:

- We decided to forego a user authentication option so as to gain users' trust by storing as little data as possible. We want to educate users about their data, not become another corporate blackbox. As such, we will discard all data used as soon as it has been through the Parser and the Analyzer. However, users can export their final summary reports once it's loaded, but upon exiting the page, the data will be lost. The users are responsible for maintaining their own data that they choose to upload onto the website.
- If time permits, we will have a required login and authentication feature for every user. This way, users can go back and see a history log of their past uploads and keep track of simple metadata relating to their upload (explained more below).

User Data Storage

<u>How much of the user's data will Verse be saving?</u>

Potential Solutions:

1) Save nothing; erase every single datum upon program completion
2) Save summary results only and throw away all raw data upon exiting the program
3) **Save summary results only and throw away all raw data except name of files upon exiting the program (time permitting)**
4) Keep all raw data that the user submits into Verse

Decision:

- We decided to save only the summary results from the raw data information along with the name of the files that the user submitted into Verse. This way, the user can recall the summary information if they wanted to as well as remind themselves which files they submitted for Verse to analyze. If we threw all data away once the user exits the program, if they wanted to access the summary results again, they would have to upload the same files and wait for the analysis to complete again, which would be a major hassle for the user, since a lot of the data dumps accumulated by major companies carry many gigabytes worth of data. As a result, analyzing the data over and over again to see the same results would be a huge waste of time on the user's end and resources on our end. On the other hand, if we saved all the information that the user submitted into the website, not only would our database be overloaded by huge amounts of information, but it would be extremely hypocritical of us to store information on our users when this site was made specifically to bring attention to the user about the vast number of sensitive information major companies have on them. If we saved their information, we would be doing the exact opposite of what we created this program to warn them against. So, if time allows, we can allow users to create an account to save their summary results and files submitted to save them time if and when they want to access the information again.

Non-Functional Issues:

Language and Framework

What language and framework would suit our development goals the most?

Potential Solutions:

1) Java & Spring Boot
2) PHP & Laravel
3) Javascript & Node.js
4) **Python & Django**

Decision:

- Since Python is one of the most used languages and one of the easiest for all team members, we have decided to write our backend with it. It natively integrates with MySQL so we also have a great contender for a database. And, since Django is one of the top frameworks for Python, we have decided to use it as our web framework. Django has facilitates quick development time and is useful because its lightweight server allows developers to quickly test changes in a non-live environment.

Hosting

How should we host the program?

Potential Solutions:

1) **Heroku**
2) Use a teammate’s computer as a server
3) Firebase
4) Github Pages

Decision:

- Heroku seems to be the easiest to set up for hosting our website. There seems to be no downsides compared to the other solutions as well. If we wanted to use our teammates computer then there could be many technical problems that we come across that would be taken care of easily if we used a hosting service. Since we also need the user to input files which changes our output, we cannot use Github due to it's static nature. Last but not least, it is more complex to build an API with Firebase, and while possible, it will impact app performance.

Data Storage

Which service will we use to store the data once it is parsed?

Potential Solutions:

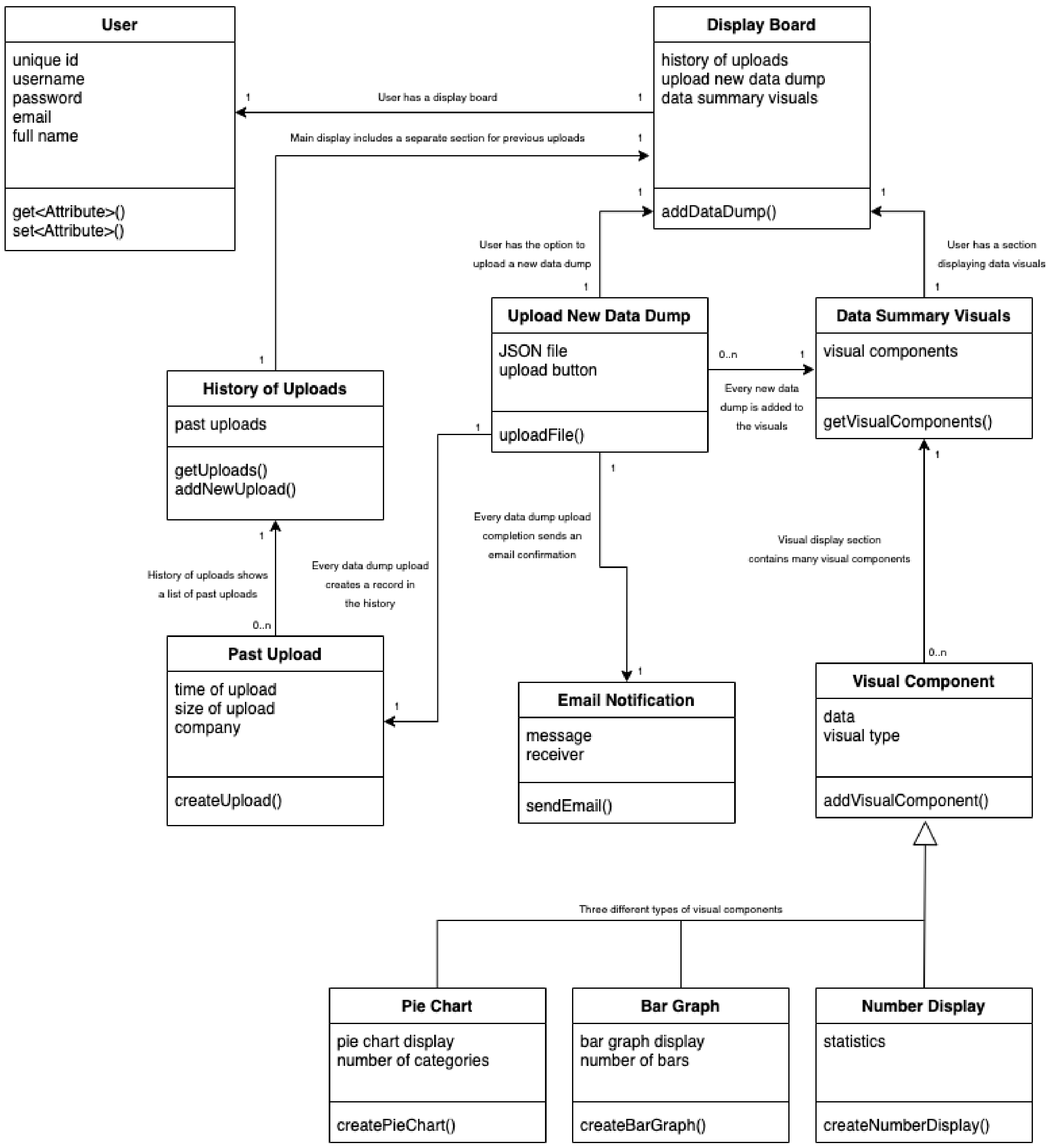
1) **MySQL**
2) MongoDB
3) Redis
4) Oracle XE

Decision:

- Since we are using Python, MongoDB is out of the question since it only runs on C/C++ and Java. This also rules out Oracle since it only works with Java. Since our group has the most experience with SQL, we would be wasting valuable time trying to learn how to use Redis and its many different components. Redis also has several different forms in Python that we'd need to go through and test with. SQL is also extremely reliable when it comes to Python implementation. Thus, it is most logical to choose MySQL as our database for Verse.

Design Details

Class Level Design:



Class and Class Interaction Descriptions

<u>User</u>

- Users will have to register in order to use Verse
- The user registration process takes in an email, username, password, and full name
- A unique id is assigned to the user
- Each user will have one display board which will hold all the information important to the user and application
- Users will have a history of their previous uploads

<u>Display Board</u>

- The display board will hold everything pertinent to the user and web application
- The upload history, option to upload a new data dump, and the section that shows a visual summary of the uploaded data can be accessed through the display board
- Every user has a display board

<u>History of Uploads</u>

- The history of uploads class will hold a list of all past uploads
- The list will be ordered by time uploaded
- When a new upload is complete, a past upload class will be created and will be added to the list
- The history of uploads can be accessed through the display board
- Every user has a history
- The upload history can be cleared by the user

<u>Past Uploads</u>

- Each past upload will show up in the history of uploads list
- Past uploads will only save the time of upload, size of data dump, and the company in which the data came from
- The minimal amount of data preserved will maximize the user’s privacy
- An upload record will be created every time a user uploads a new data dump

Upload New Data Dump

- Users have the option to upload a new data dump
- This class can be accessed through a visual button on the display board
- Uploading a new data dump requires a separate web page
- The company in which the data is uploaded from must be specified
- Every data dump upload will create a past upload object which is added to the upload history
- The data dumps must be uploaded in a JSON format
- The process of uploading a data dump uses one of the various data parsers specific to each company
- Once the upload is complete, the parsed data will be sent to the data summary visuals and also back to the display board in order to create the summarized visuals

Email Notification

- In the case that uploads take a long time, an email notification will be sent once the data dump upload is complete
- The notification is initiated by the completion of an upload
- An email is sent to the user’s registered email as confirmation

Data Summary Visuals

- The data summary visuals will be created from the parsed data
- There will be one section for all of the visuals regarding a data upload
- Users that upload multiple data dumps per session will have all the parsed data compared on one data summary visuals section
- This section is included in the display board
- The data summary visuals section is comprised of many visual components, each displaying a specific part of the parsed data

Visual Component

- There are many visual components within the data summary visuals section
- Each visual component is specific to a section in the parsed data
- There are several different types of visual components: pie chart, bar graph, and number display

- For every data dump upload, several visual components will be created in order to display the JSON data in a more visually friendly format

Pie Chart

- Pie charts are a type of visual component
- A pie chart will be used to display any information in which categories make up a whole
- The parsed data will be used in order to create the chart

Bar Graph

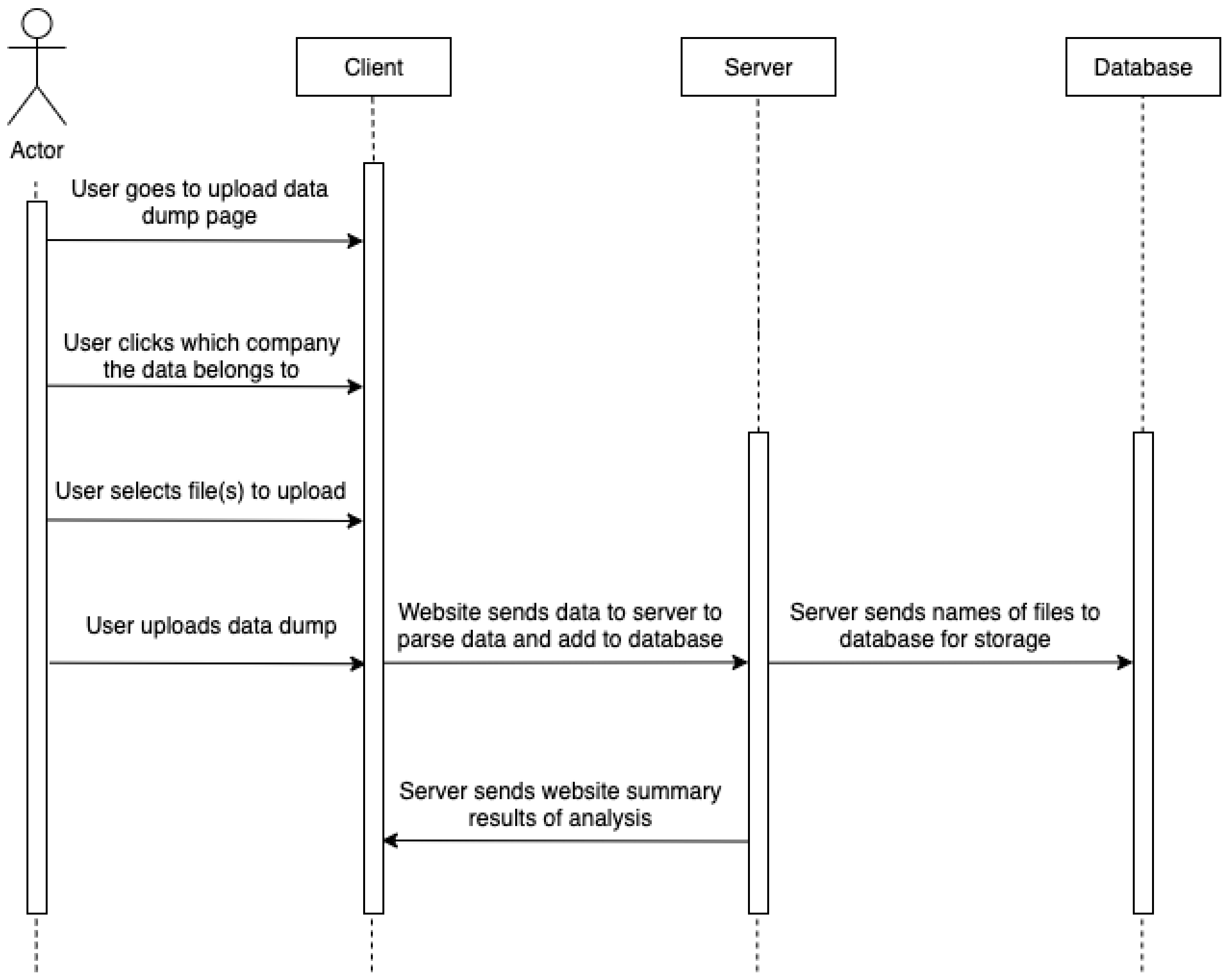
- Bar graphs are a type of visual component
- A bar graph will be used to display information in which categories are compared
- The parsed data will be used in order to create the graph

Number Display

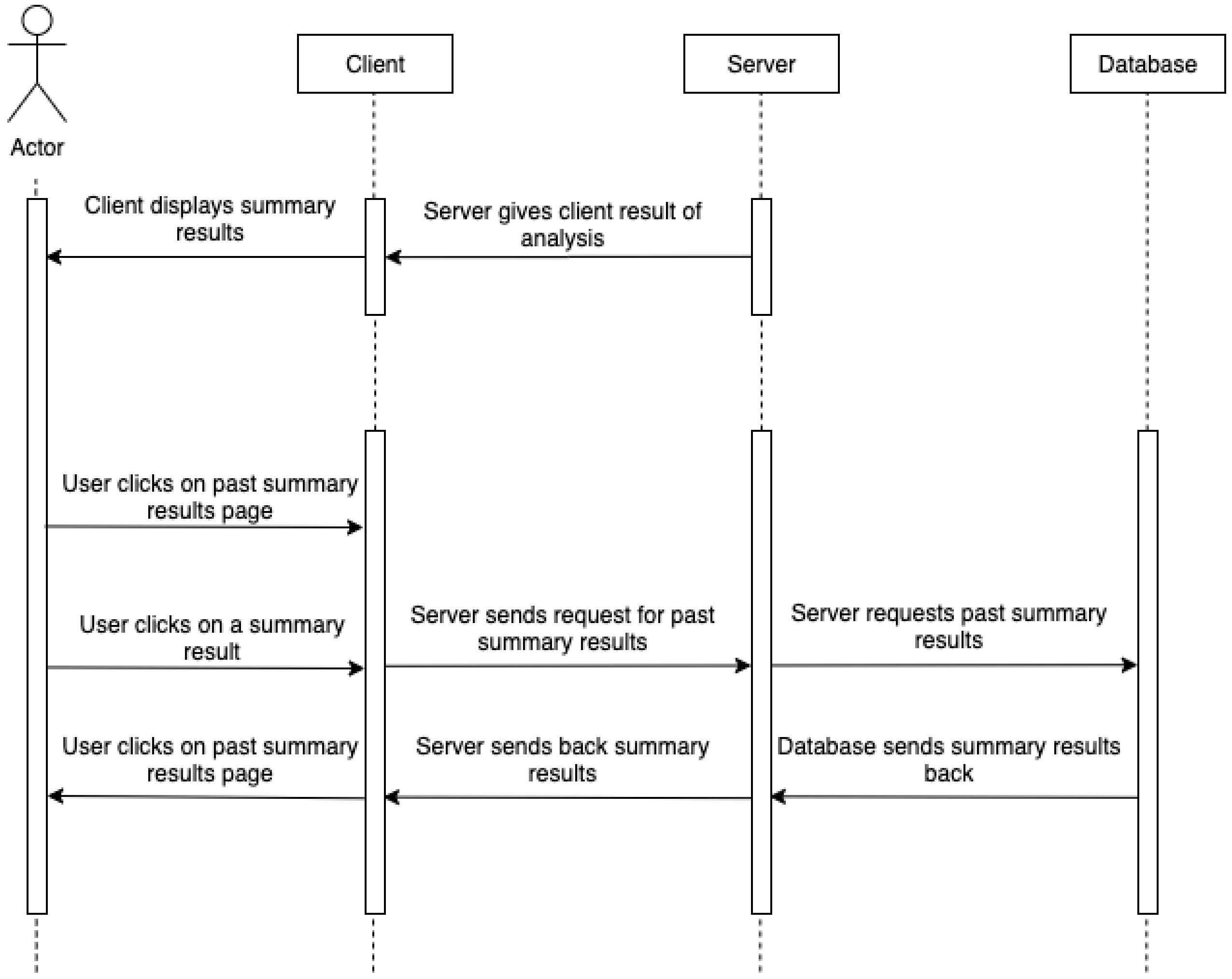
- Number displays are a type of visual component
- A number display will be used to display simple numerical data, such as the size of the upload
- The parsed data will be used in order to create the graph

Activity Diagrams

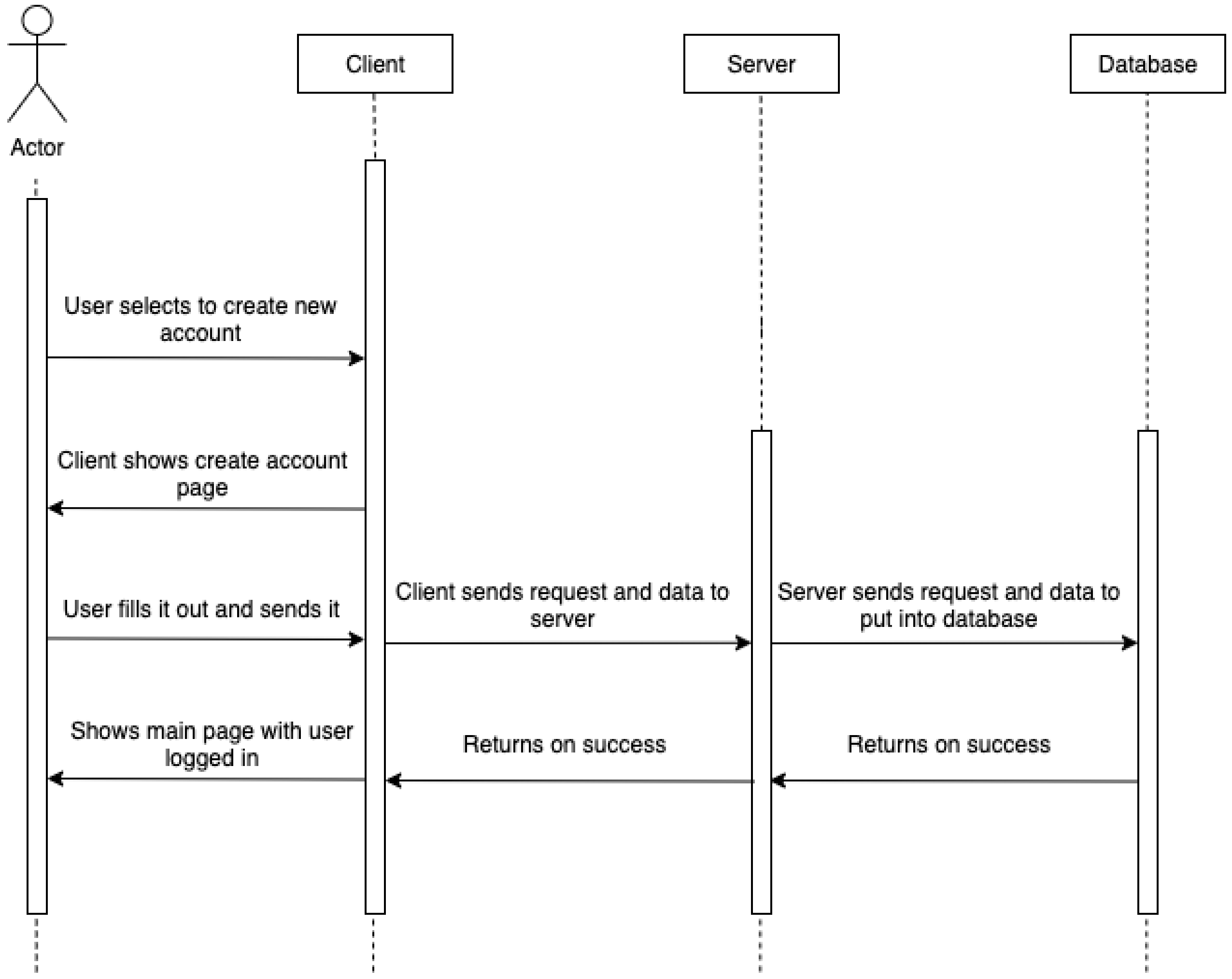
Sequence Diagram for Inputting Files


Actor
Client
Server
Database
User goes to upload data dump page
User clicks which company the data belongs to
User selects file(s) to upload
User uploads data dump
Website sends data to server to parse data and add to database
Server sends names of files to database for storage
Server sends website summary results of analysis

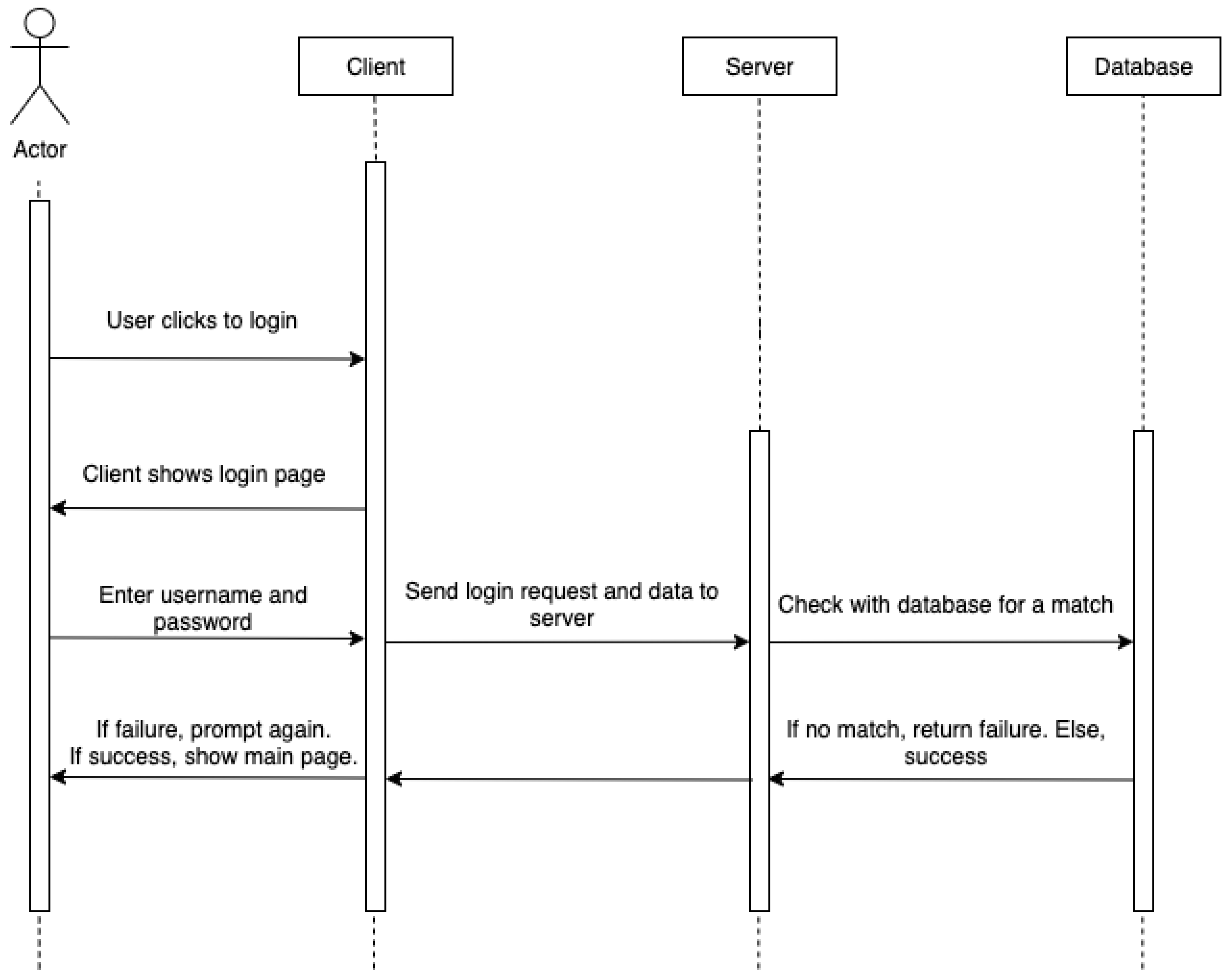
Sequence Diagram for Looking at Data Summary Results


Actor
Client
Server
Database
Client displays summary results
Server gives client result of analysis
User clicks on past summary results page
User clicks on a summary result
Server sends request for past summary results
Server requests past summary results
User clicks on past summary results page
Server sends back summary results
Database sends summary results back

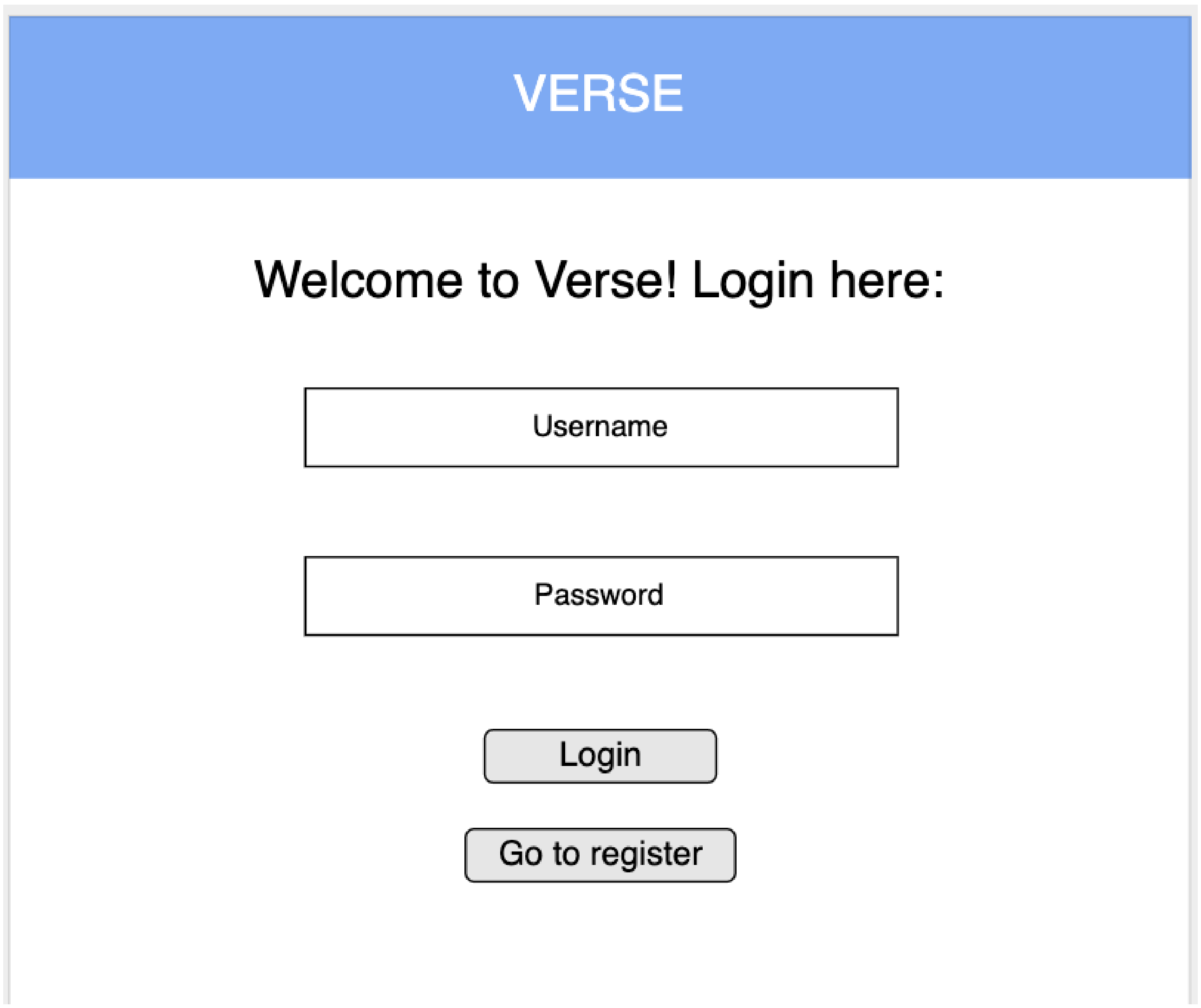
Sequence Diagram for Creating New Account


Actor
Client
Server
Database
User selects to create new account
Client shows create account page
User fills it out and sends it
Client sends request and data to server
Server sends request and data to put into database
Shows main page with user logged in
Returns on success
Returns on success

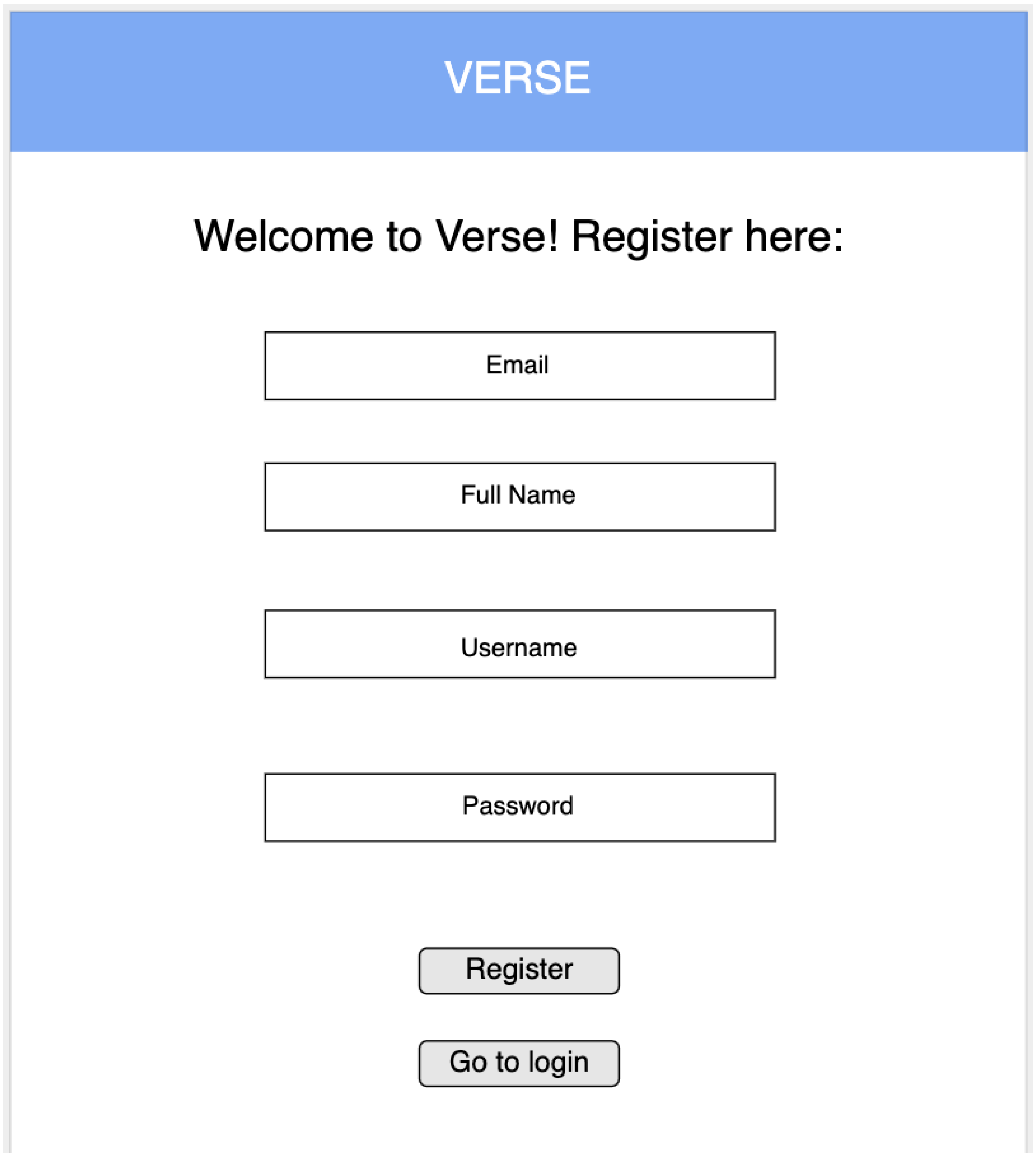
Sequence Diagram for Logging in


Actor
Client
Server
Database
User clicks to login
Client shows login page
Enter username and password
Send login request and data to server
Check with database for a match
If failure, prompt again. If success, show main page.
If no match, return failure. Else, success

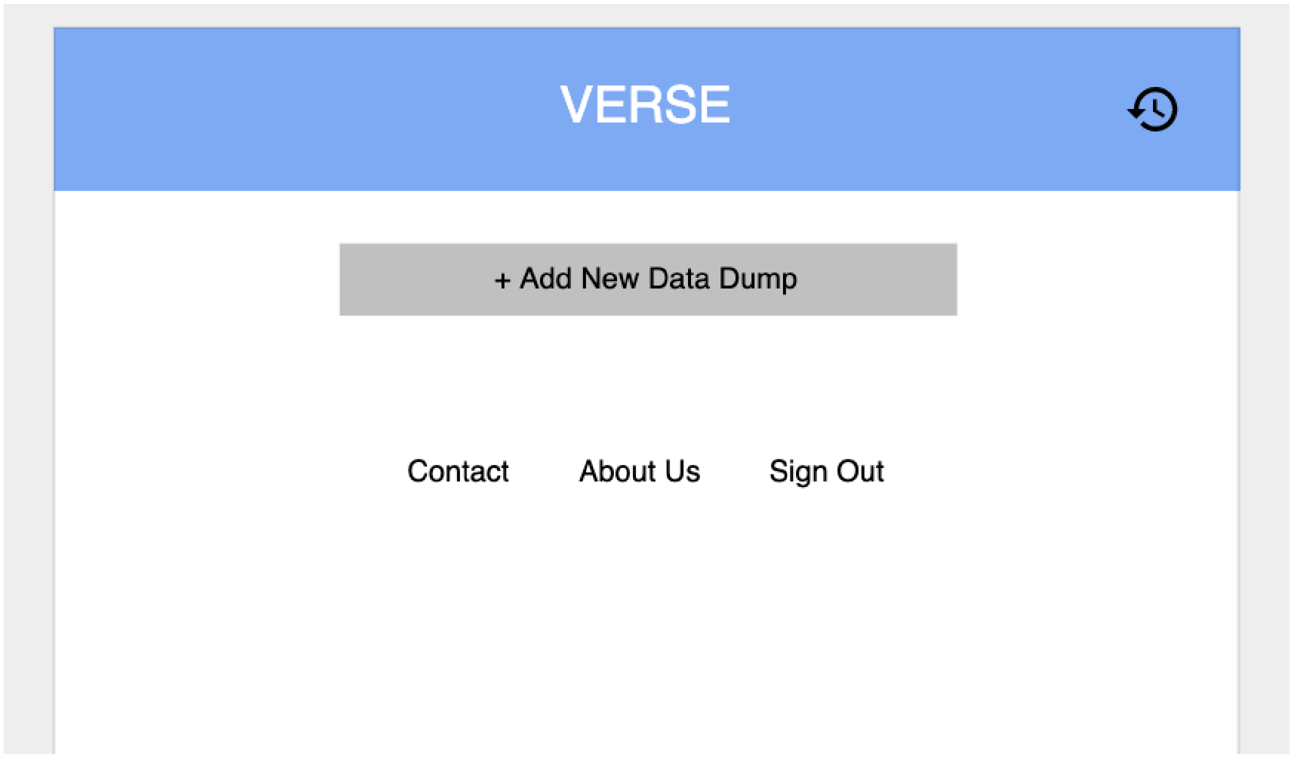
UI Mockups



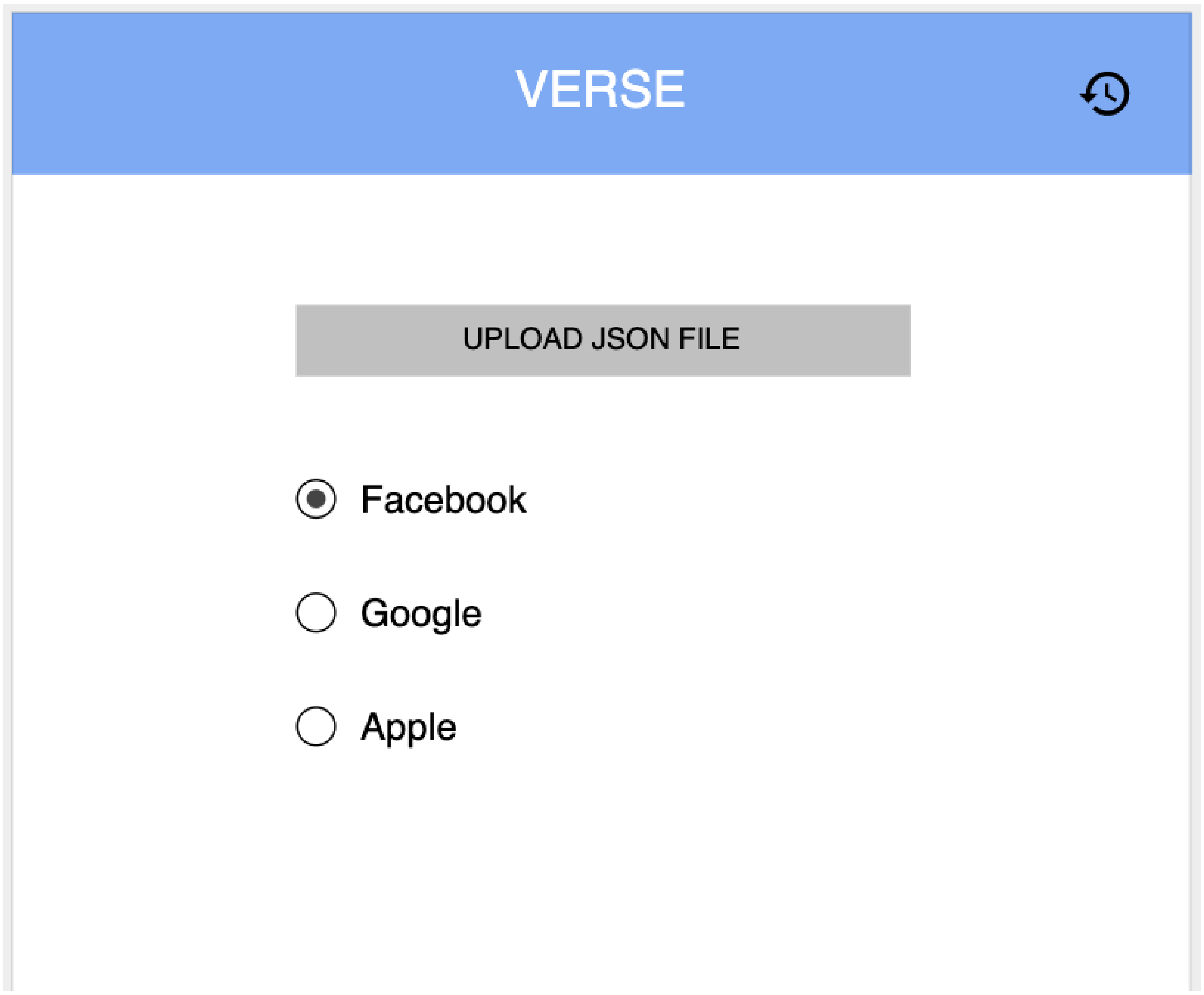
This is the mockup for the login page. For users that already have an account, they can log in here.



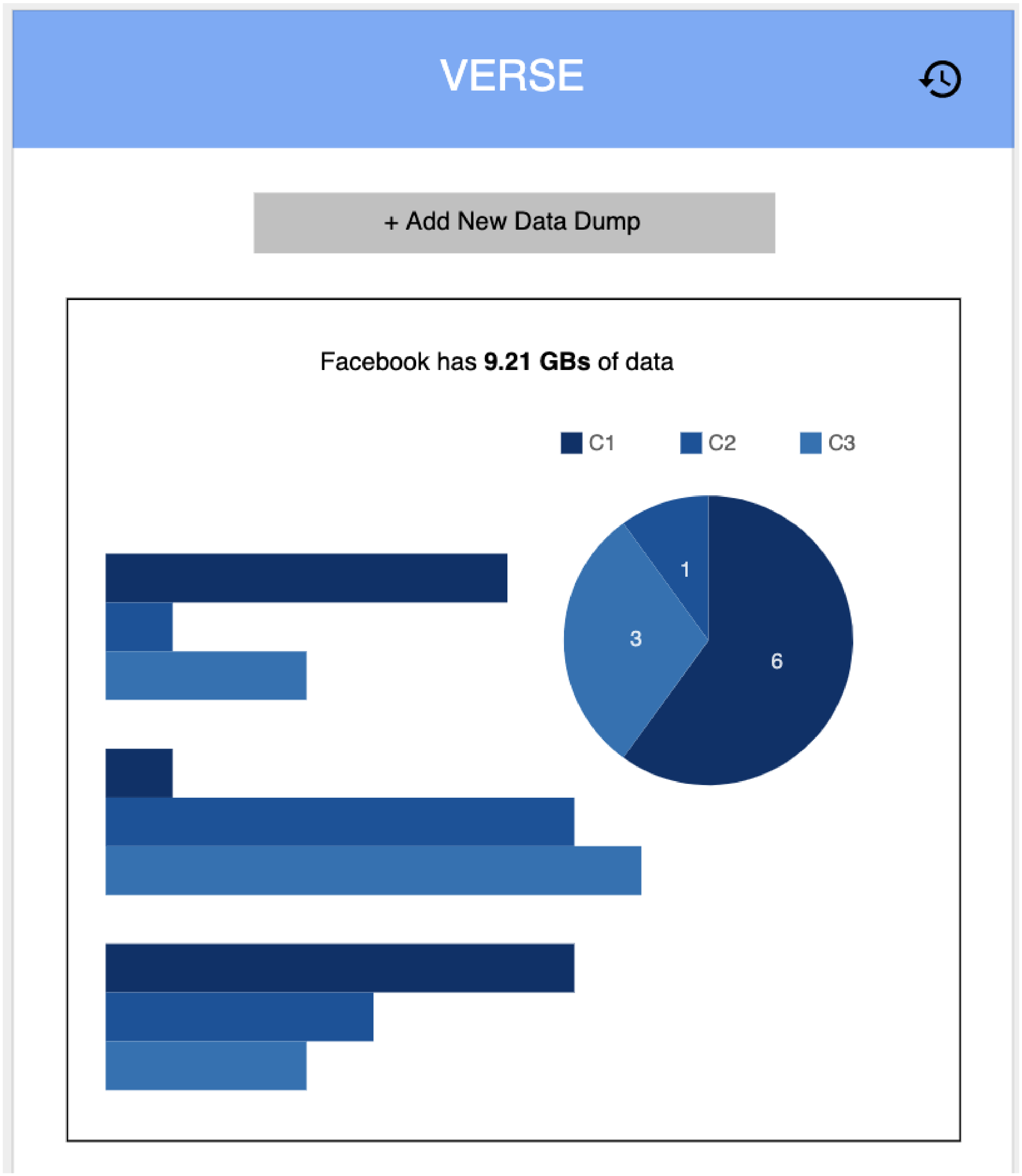
Alternatively, users can register for an account on this page. This will register them for our service and allow them to log in.



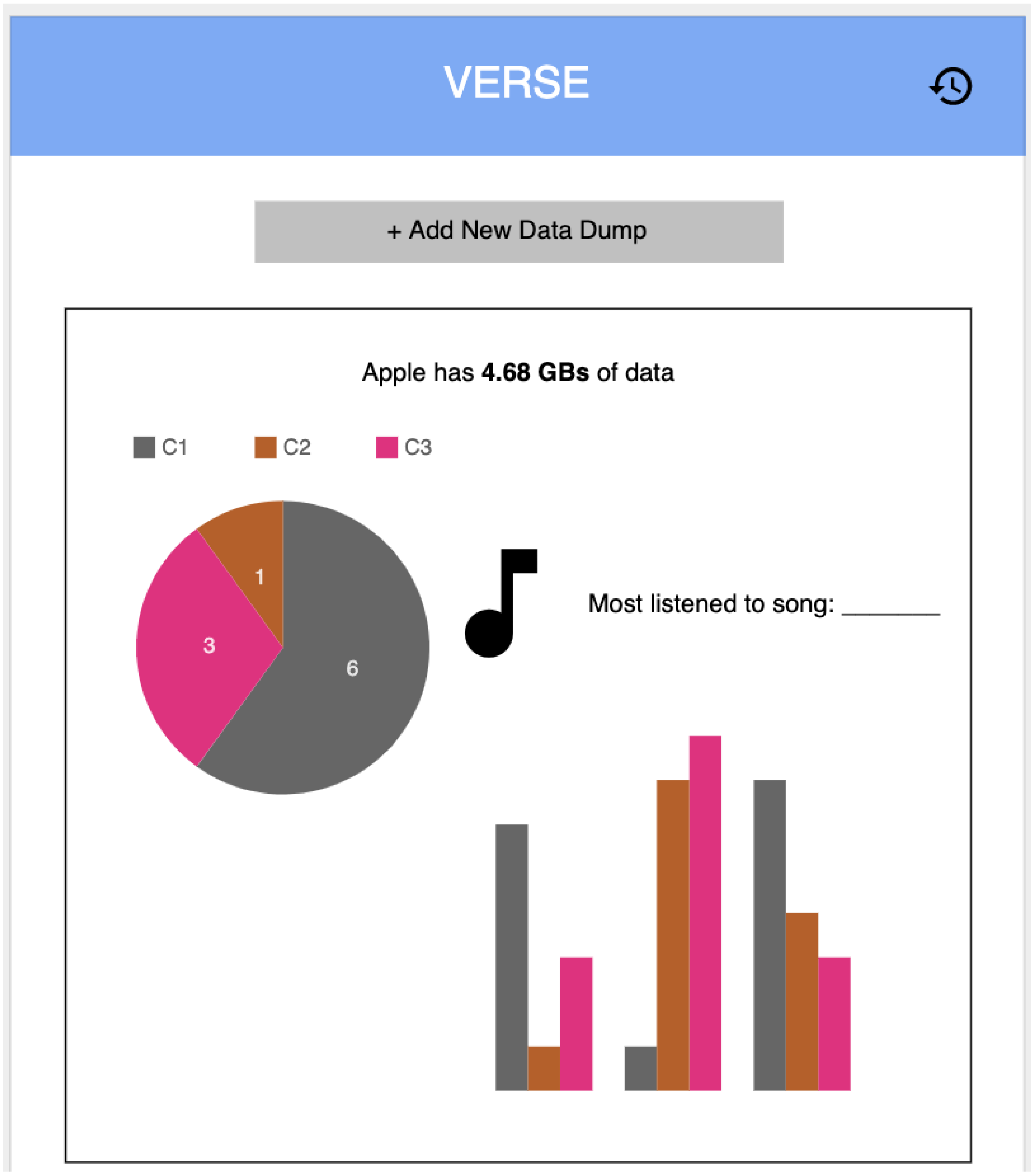
This will be the main page before any data dumps are uploaded. Since there is nothing uploaded, there is no data for our service to parse. Therefore, no visuals will be displayed. This will change once a data dump is uploaded.



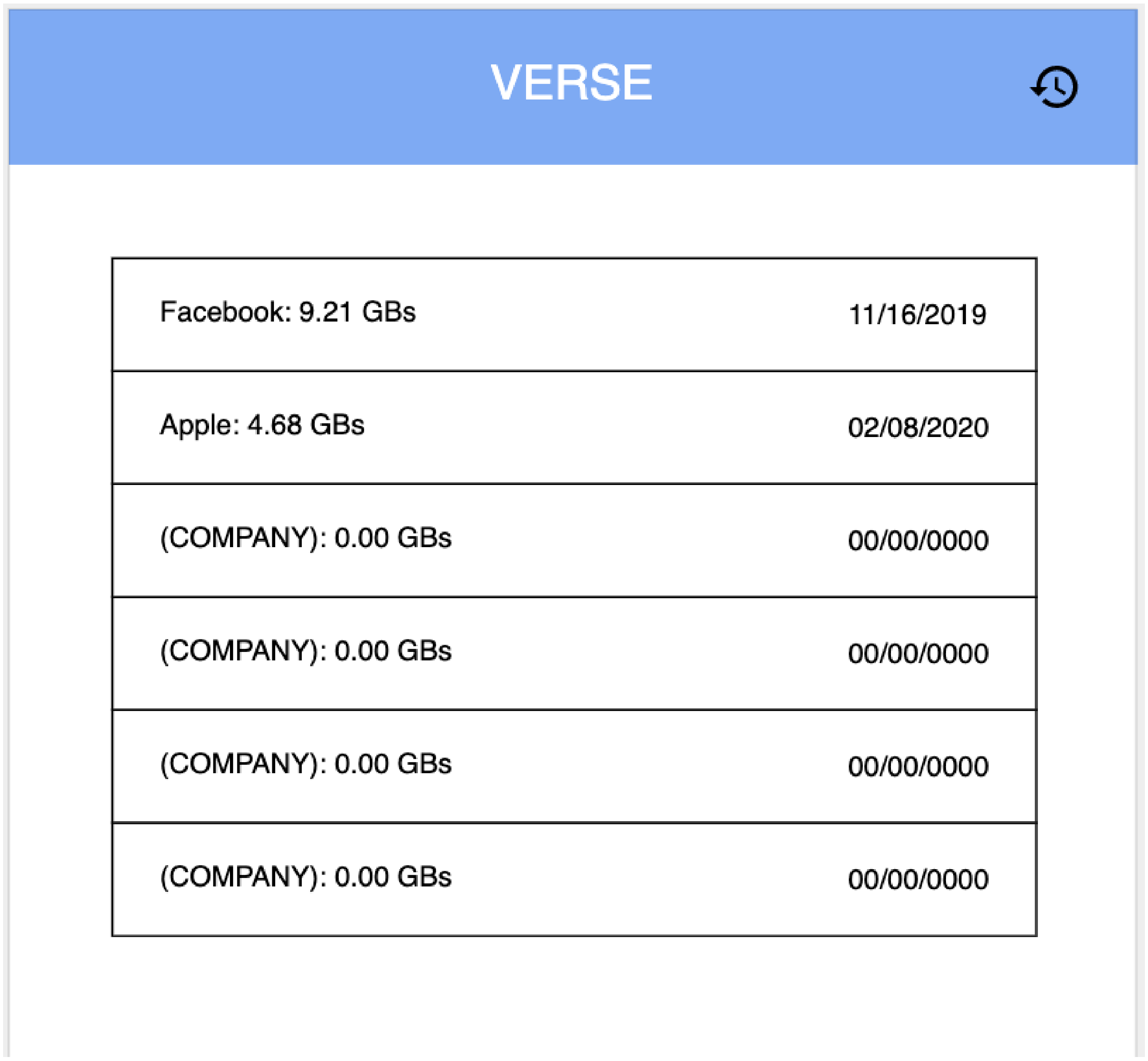
After clicking '+ Add a new Data Dump', the user will be directed to this page. Here, the user can upload a JSON file and specify which company the file came from. This will allow the data parser to sort through all of the data and return a summarized JSON file that can be visualized.



This is the main page with a data dump uploaded. The visuals will vary based on the data uploaded and the amount of data in the JSON file. These visual charts are placeholders, and the actual displays will be different.


VERSE
+ Add New Data Dump
Apple has **4.68 GBs** of data
C1
C2
C3
1
3
6
Most listened to song: ________

This is an alternative visual display screen. This demonstrates the possible differences between visuals based on where the data dumps originate from.



VERSE

Facebook: 9.21 GBs	11/16/2019
Apple: 4.68 GBs	02/08/2020
(COMPANY): 0.00 GBs	00/00/0000
(COMPANY): 0.00 GBs	00/00/0000
(COMPANY): 0.00 GBs	00/00/0000
(COMPANY): 0.00 GBs	00/00/0000

This is a page for the user's history of uploads. Our web app will only record the metadata of the data dumps, such as the company it came from, the size of the upload, and the time uploaded. In the list, all the upload records past the second row are placeholders that demonstrate what the list would look like with more uploads. This page can be reached by clicking the history icon in the navigation bar.