

WIF3002 Software Process and Quality

Software Requirements Specification

For

My Covid View

| **Name** | **Matric No.** |
| --- | --- |
| Kang Zhi Hui | U2005282/1 |
| Goh Kang Yue | U2005257/1 |
| Cheng Mou Yuan | U2005361/1 |
| Tan Hou Ren | S2030685/1 |
| Lim Jing Ying | U2005397/1 |
| Chue Kai Ze | U2005360/1 |
| Chong Chien Wang | S2030951/1 |
| Lim Zi Xuan | U2005290/1 |
| Muhammad Ariff Haikal bin Hurairy | 17202990/2 |
| Mohd Izzatul Fahmi bin Kassim | 17204883/2 |
| Liew Pei Mun | U2005268/1 |
| Pua Jing Yi | U2005396/1 |
| Jocelyn Foo Xiao Yun | S2032727/1 |
| Lee Yi Wan | U2005309/1 |
| Tan Chiao Yun | U2004576/1 |

Lecturer : Dr. Raja Jamilah Binti Raja Yusof

# 

[**STAKEHOLDERS 4**](#_qzlmhdf9rf9k)

[**INTRODUCTION 4**](#_e7btb8ca15sm)

[**BACKGROUND 6**](#_x54mpdyoyzdh)

[1.1 Organisational Structure 6](#_4zrizdv74wam)

[**SYSTEM OBJECTIVE 6**](#_psygevgp3j5g)

[2.1 Business Goal 6](#_ipmmcqa6e4eb)

[2.2 Problem to be solved 6](#_15csa15o9j4s)

[2.3 System Constraint 7](#_94e3dbp5h9yh)

[**REQUIREMENTS 8**](#_hzvb8bi1d7y7)

[3.1 Quality Requirements in Natural Language 8](#_acqnqf3ozawo)

[3.1.1 Usability 8](#_a35k9b6mum9k)

[3.1.2 Portability 8](#_lg5ngzgszzdw)

[3.1.3 Maintainability 8](#_xk3fr7ycssed)

[3.1.4 Reliability 9](#_it49i5lqmyac)

[3.2 List of Functional Requirements in Natural Language 9](#_1lfvb6zcgt24)

[**OBJECT-ORIENTED UML MODELLING 11**](#_ja3wsg3tot04)

[4.1 UML Use Case Diagram 11](#_7gb6migk5rg6)

[4.2 UML Use Case Description 12](#_heac1fccnoh4)

[UC - 1 import data 12](#_tzeegel3uz3i)

[UC - 2 export data 14](#_vaxryhaeevg1)

[UC - 3 clean and process data 16](#_1lbzm97tibu1)

[UC - 4 view visualized data in web app 18](#_9rbf7t96mmyn)

[UC - 5 compare visualized data 20](#_3eyied4brkl3)

[UC - 6 delete data to be visualized 22](#_86dxfoync4uq)

[UC - 7 update data to be visualized 24](#_6cfjvbishj4e)

[UC - 8 zoom in and out data visualization 25](#_ylyy96gsudq)

[UC - 9 filter and sort data 26](#_8qd56lt2owe5)

[UC - 10 change visualization technique 27](#_kkmhmqg6yn8k)

[UC - 11 change data to be visualized 29](#_d4rwpmfqmnjn)

[UC - 12 change data visualization colour 31](#_sogl5sn8f73m)

[UC - 13 view visualized data in iOS and Android apps 33](#_t0y3up4tyh6d)

[4.3 UML Class Diagram 35](#_d4cstuijxfwj)

[4.4 UML Sequence Diagram 35](#_8dwoi126e66e)

[4.5 UML Activity Diagram 42](#_hwxd98h6rg5h)

[4.6 UML Component Diagram 55](#_1i14wuw2gz52)

[4.7 UML Deployment Diagram 56](#_oh2c37e75hz4)

[**USER INTERFACE REQUIREMENTS 57**](#_t8xaeilzkw0c)

[5.1 Mockup screen (Wireframe Version 1) 57](#_rnbg0263p019)

[5.1.1 Web View(Monitor 1920px wide) 57](#_5wc70xi2q0k0)

[5.1.2 Web View(Laptop 1024px wide) 60](#_br2juydyby5g)

[5.1.3 Web View(Tablet 768px wide) 63](#_7zovlrpvbkf3)

[5.1.4 Web View(Mobile 320px wide) 66](#_xvwuwxf7rc03)

[5.2 Mockup screen(Wireframe Version 2) 69](#_pvywpuflyfic)

[5.2.1 Web View(Monitor 1920px wide) 69](#_gmwm7qm64qir)

[5.2.2 Web View(Laptop 1024px wide) 69](#_83yf15miaht5)

[5.2.3 Web View(Tablet 768px wide) 70](#_dtgicrpschnd)

[5.2.4 Web View(Mobile 320px wide) 70](#_pzs3hivkzeu8)

[5.3 Application navigation (flow of the system) 70](#_drqlpdk9ws5s)

# 

# STAKEHOLDERS

* Business professionals
* Data analysts
* Researchers
* Educators

# INTRODUCTION

This report details the software requirements for the development of a data visualization application. The application is designed to provide users with powerful insights into their data through a variety of visualization techniques, thereby catering to a diverse user base including business professionals, data analysts, researchers, and educators. The application is compatible with both iOS and Android platforms, as well as the web browsers.

Data visualization is a graphical representation of data and information. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. In the context of this project, the application incorporates a variety of visualization techniques, such as bar charts, line charts, scatter plots, pie charts, histograms, box plots, treemaps, sunburst charts, infographics, heat maps, bubble charts, dendrograms, scorecards, KPI dashboards, and balanced scorecards. These techniques are selected based on the specific needs and goals of each visualization, as well as the preferences and needs of the application's target audience.

The software requirements for this application were elicited and analyzed using the Design Thinking methodology, which included stages such as Empathy and Define, Ideation, and Design. This involved tasks such as software requirement elicitation and analysis, sketching and structuring the idea of how to design the app, laying out the conceptual design, wireframing the interface, data processing and cleaning, and user interface design.

The report begins by identifying the stakeholders involved in the project, followed by a background section that provides an overview of the organizational structure. The system objective is then discussed, outlining the business goal, the problem to be solved, and any system constraints.

The core of the report lies in the requirements section, which details the quality requirements in natural language, focusing on aspects such as usability, portability, maintainability, and reliability. It also lists the functional requirements of the application.

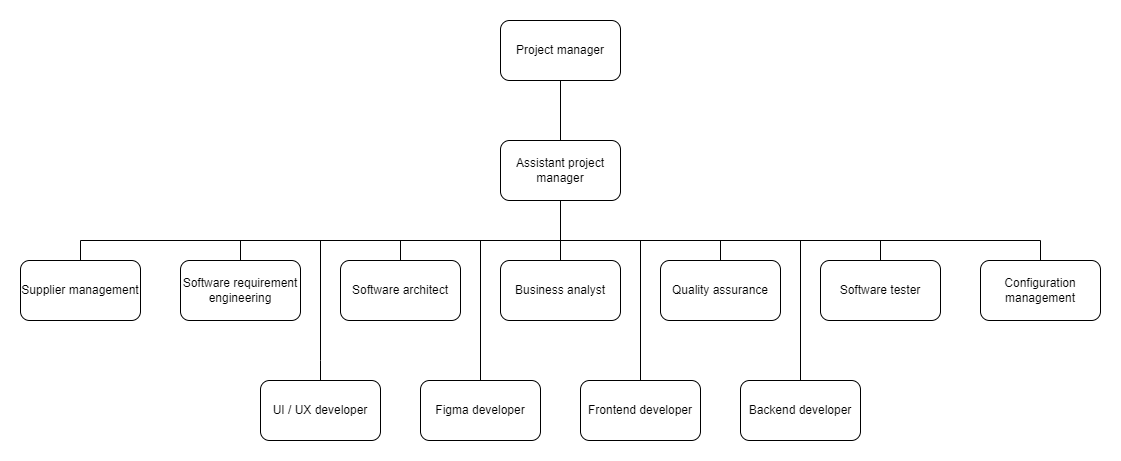
To provide a visual representation of the system, the report includes an Object-Oriented UML Modelling section. This section presents various UML diagrams, including Use Case, Class, Sequence, Activity, Component, and Deployment diagrams, to illustrate the system's structure, behavior, and interactions.

Finally, the report discusses the user interface requirements, providing mockup screens and detailing the application's navigation flow. This section aims to ensure that the application is user-friendly and meets the needs of its intended audience.

Throughout the report, the focus is on defining clear, concise, and complete requirements to guide the development process and ensure the successful realization of the data visualization application.

# BACKGROUND

## **1.1 Organisational Structure**



# SYSTEM OBJECTIVE

## **2.1 Business Goal**

* To support a range of different visualization goals.
* To ease data interpretation for users from different aspects to meet their specific needs and goals.
* To implement a robust data processing and visualization engine that can handle large datasets.
* To support cross compatibility such as in web and mobile apps.
* To optimize the performance and the speed of data processing and data interpretation.
* To improve the first time user experience on the visualization app.

## **2.2 Problem to be solved**

* It is hard for people to interpret a huge amount of data by conducting the analysis process one by one or only a few graphs.
  + There are some datasets consisting of many rows and columns and it needs time to process and clean the data before visualising the data. It is time consuming and decreases the efficiency of the work.
* There are limited visualization apps that apply different visualization techniques for users to interpret the data from different aspects.
  + As we know, it is easy to interpret the datasets by visualizations as it is easy to view instead of a string of words, rows and columns. By interpreting different visualizations, we can conclude a good and high accuracy of the result of the datasets.
* The users claim that there are some visualization apps in the market that are difficult to master as the app design is complicated when they are the first time users.
  + Although there are apps provided in the market for different visualization techniques, it is not user friendly to the first time users. They need to go through some tutorial videos on how to use the apps before they can become completely familiar with it.

## **2.3 System Constraint**

* The system should be able to handle more than 1000 login sessions at the same time.
* The system should provide a responsive UI interface for different screen resolutions for mobile and web applications.
* The system should not allow the users to store the data uploaded in its server.
* The system should be able to process more than 1 dataset at the same.
* The system should be compatible for various types of operating systems in mobile and web applications.

# REQUIREMENTS

## **3.1 Quality Requirements in Natural Language**

### 3.1.1 Usability

Usability refers to the degree to which a product or system can be used by specified users to achieve specific goals. As a data visualisation system, it should be intuitive and easy to be learnt and operated to get the output easily.

**NFR - 001:** The users shall be able to navigate easily across different pages seamlessly by clicking the relevant icons / buttons / links.

**NFR - 002:** The users shall be able to learn how to use the application such as import, export and clear data, view, update, and delete the visualised data and zoom in and out the graph within 2 hours.

**NFR - 003:**  The user interface of this application is intuitive, attractive, and usable.

### 3.1.2 Portability

Portability is the effort to transfer the program from one hardware and/or software system environment to another. The data visualisation system created should be portable across different OS platforms and different devices.

**NFR - 004:** The user is capable of running the application on different operating systems such as iOS and Android.

**NFR - 005:** The user is capable of running the application on different devices such as a mobile phone, a tablet, a personal computer (PC) or a laptop.

### 3.1.3 Maintainability

Maintainability is the effort required to locate and fix an error in a program. In the data visualisation system, we should use a modular and scalable architecture to design and develop it.

**NFR - 006:** The application shall separate the graph visualisation such as line chart, area chart, and scatter plot into three different modules because they serve for three different purposes.

**NFR - 007:** The application shall have comprehensive documentation such as UML diagram, functional requirements, how it works and how it is developed to ease future changes.

### 3.1.4 Reliability

Reliability is the extent to which a program can perform its intended function with required precision. In the data visualisation system, it should be tested rigorously to identify and fix the bugs to make sure the application works under different conditions.

**NFR - 008:** With the presence of the Internet, the application is available 24/7 for the user with all its functional features such as import, export and clear data, view, update, and delete the visualised data and zoom in and out the graph.

**NFR - 009:** If there is a new version of the application, outdated versions should be supported by the system.

## **3.2 List of Functional Requirements in Natural Language**

**FR - 001:** The system shall allow the actor to import data from various sources (e.g., CSV files, APIs).

**FR - 002:** The system shall allow actors to export visualizations into various formats (like PNG, JPEG, PDF, etc) for use in reports or presentations.

**FR - 003:** The system shall allow actors to clean data by performing operations such as handling missing data, removing duplicate rows, converting wrong data type, standardizing or normalizing numeric data, etc.

**FR - 004:** The system shall allow actors to develop various data visualization techniques across various web browsers and screen sizes.

**FR - 005:** The system shall allow actors to view visualized data within a web application.

**FR - 006:** The system shall allow actors to compare different datasets or data points within the same visualization for comparative analysis.

**FR - 007:** The system shall allow actors to remove specific datasets or data points from their visualizations.

**FR - 008:** The system shall allow the actor to add new data to their existing visualizations.

**FR - 009:** The system shall allow the actor to magnify or reduce their view of a data visualization.

**FR - 010:** The system shall allow the actor to filter or sort data in a specific order before or after visualization.

**FR - 011:** The system shall allow the actor to conveniently change the method used to visualize their data, enabling them to switch between different visualization techniques.

**FR - 012:** The system shall enable the actor to select or deselect different datasets or data points for visualization.

**FR - 013:** The system shall allow the actor to customize the color scheme of their data visualization for better readability or aesthetics.

**FR - 014:** The system shall enable the actor to access and view visualized data on both iOS and Android platforms through dedicated mobile applications.

# OBJECT-ORIENTED UML MODELLING

## **4.1 UML Use Case Diagram**

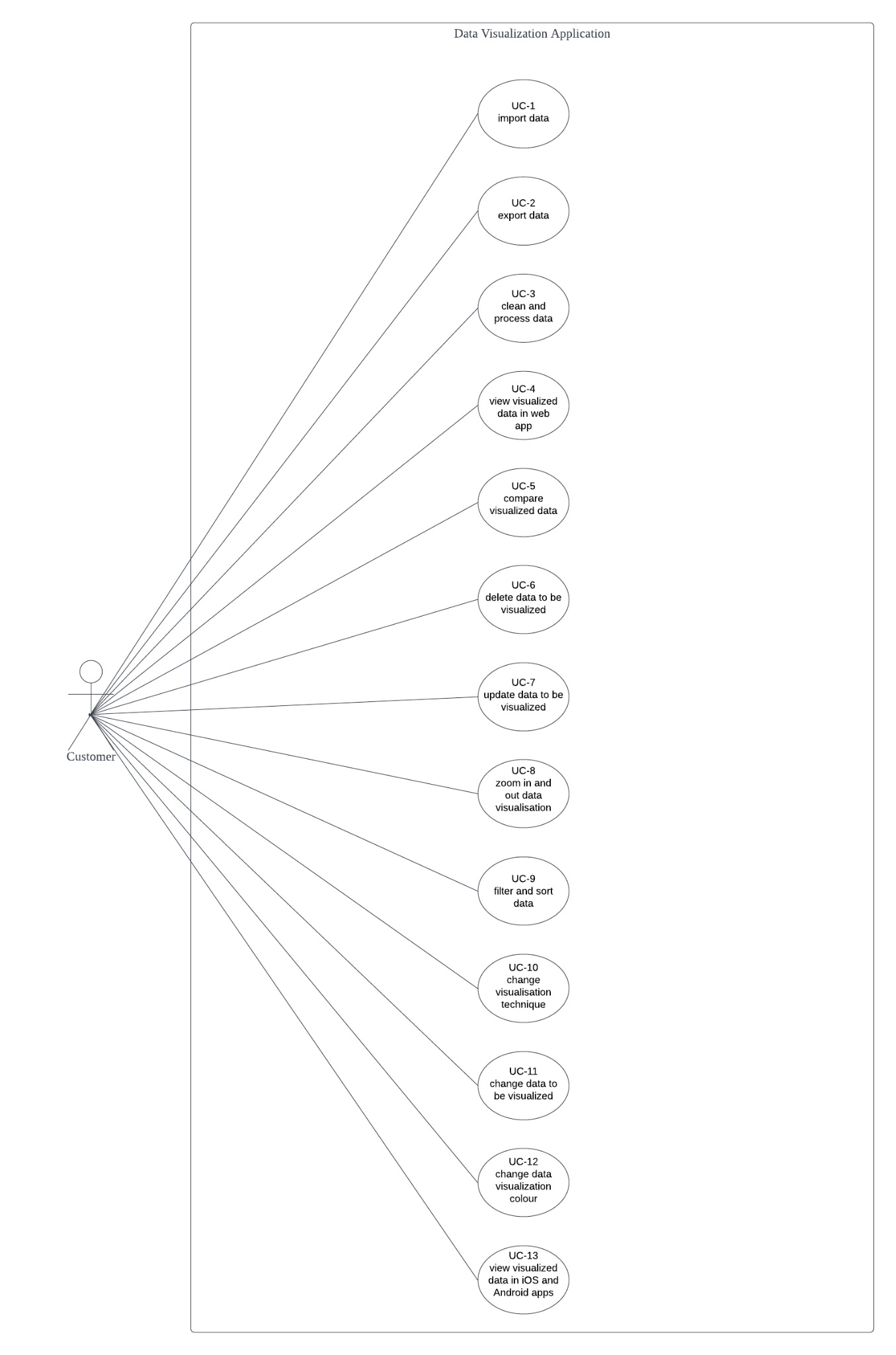


Figure 4.1.1: Use Case Diagram

## **4.2 UML Use Case Description**

### **UC - 1 import data**

| Use Case Name | import data |
| --- | --- |
| Use Case ID | UC-1 |
| Description | The system shall allow the actor to import data from various sources (e.g., CSV files, APIs). |
| Priority | High |
| Actor(s) | Customer |
| Triggering Event | The actor clicks on the “Import Data” button. |
| Pre-condition | 1. The actor has valid access to the system. 2. The actor has the necessary permissions to import data. 3. The data source (CSV files, APIs, etc.) is available and accessible. 4. The format of the data is supported by the system. |
| Post-condition | 1. The data from the chosen source has been imported into the system. 2. The actor is informed of the successful data import or of any issues that occurred. |
| Flow of Events | 1. The actor logs into the system. 2. The actor navigates to the data import section. 3. The actor clicks on the “Import Data” button. 4. The actor chooses the “Import Data” option. 5. The system presents the actor with a selection of data sources. 6. The actor selects the desired data source. 7. The system prompts the actor for file or API details. 8. The system validates the data source. 9. The system imports the data. 10. The system notifies the actor. |
| Alternative Flow | N/A |
| Exception Flow -  **@Step 8:** The actor provides an unsupported file format. | The system displays an error message. |
| Exception Flow -  **@Step 8:** The actor provides invalid API details. | The system displays an error message. |
| Exception Flow -  **@Step 9:** Data import process fails due to system issues. | The system displays an error message. |

Table 4.2.1: UC - 1 import data

### 

### **UC - 2 export data**

| Use Case Name | export data |
| --- | --- |
| Use Case ID | UC-2 |
| Description | The system shall allow actors to export visualizations into various formats (like PNG, JPEG, PDF, etc) for use in reports or presentations. |
| Priority | High |
| Actor(s) | Customer |
| Triggering Event | The actor clicks on the “Export Data” button. |
| Pre-condition | 1. The actor has valid access to the system. 2. The actor has the necessary permissions to export data. 3. The visualization data is available and accessible. |
| Post-condition | 1. The visualization has been exported into the chosen format. 2. The actor is informed of the successful data export or of any issues that occurred. |
| Flow of Events | 1. The actor logs into the system. 2. The actor navigates to the data visualization section. 3. The actor clicks on the “Export Data” button. 4. The actor chooses the “Export Data” option. 5. The system presents the actor with a selection of export formats. 6. The actor selects the desired export format. 7. The system prompts the actor to confirm the export. 8. The actor confirms the export. 9. The system exports the visualization. 10. The system notifies the actor. |
| Alternative Flow | N/A |
| Exception Flow - **@Step 5:** The actor chooses an unsupported export format. | The system displays an error message. |
| Exception Flow - **@Step 9:** Visualization export process fails due to system issues. | The system displays an error message. |

Table 4.2.2: UC - 2 export data

### 

### **UC - 3 clean and process data**

| Use Case Name | clean and process data |
| --- | --- |
| Use Case ID | UC-3 |
| Description | The system shall allow actors to clean data by performing operations such as handling missing data and removing duplicate rows. |
| Priority | High |
| Actor(s) | Customer |
| Triggering Event | The actor chooses the 'Clean and Process Data' option within the system. |
| Pre-condition | 1. The actor has valid access to the system. 2. The actor has the necessary permissions to clean and process data. 3. The raw data is available and accessible. |
| Post-condition | 1. The data has been cleaned and processed as per the actor's instructions. 2. The actor is informed of the successful data cleaning and processing or of any issues that occurred. |
| Flow of Events | 1. The actor logs into the system. 2. The actor navigates to the data cleaning and processing section. 3. The actor chooses the 'Clean and Process Data' option. 4. The system presents the actor with a selection of cleaning and processing options. 5. The actor selects the desired cleaning and processing operations. 6. The system prompts the actor to confirm the operations. 7. The actor confirms the operations. 8. The system performs the operations. 9. The system notifies the actor. |
| Alternative Flow | N/A |
| Exception Flow -  **@Step 6:** The actor chooses an unsupported cleaning or processing operation. | The system displays an error message. |
| Exception Flow -  **@Step 8:** Cleaning or processing operation fails due to system issues. | The system displays an error message. |

Table 4.2.3: UC - 3 clean and process data

### 

### **UC - 4 view visualized data in web app**

| Use Case Name | view visualized data in web app |
| --- | --- |
| Use Case ID | UC-4 |
| Description | 1. The system shall allow actors to develop various data visualization techniques across various web browsers and screen sizes. 2. The system shall allow actors to view visualized data within a web application. |
| Priority | High |
| Actor(s) | Customer |
| Triggering Event | The actor navigates to the visualized data section within the web application. |
| Pre-condition | 1. The actor has valid access to the web application. 2. The actor has the necessary permissions to view visualized data. 3. The visualized data is available and accessible. |
| Post-condition | 1. The actor has viewed the visualized data within the web application. |
| Flow of Events | 1. The actor logs into the web application. 2. The actor navigates to the visualized data section. 3. The system displays the available visualizations. 4. The actor selects the desired visualization to view. 5. The system displays the visualization in the actor's web browser. |
| Alternative Flow | N/A |
| Exception Flow -  **@Step 5:** Visualization fails to load in the actor's browser. | The system displays an error message. |
| Exception Flow -  **@Step 5:** Viewing process is interrupted due to system issues. | The system displays an error message. |

Table 4.2.4: UC - 4 view visualized data in web app

### 

### **UC - 5 compare visualized data**

| Use Case Name | compare visualized data |
| --- | --- |
| Use Case ID | UC-5 |
| Description | The system shall allow actors to compare different datasets or data points within the same visualization for comparative analysis. |
| Priority | High |
| Actor(s) | Customer |
| Triggering Event | The actor chooses the 'Compare Visualized Data' option within the system. |
| Pre-condition | 1. The actor has valid access to the system. 2. The actor has the necessary permissions to compare visualized data. 3. The datasets or data points for comparison are available and accessible within the visualization. |
| Post-condition | 1. The datasets or data points have been compared within the same visualization as per the actor's instructions. 2. The actor is informed of the successful data comparison or of any issues that occurred. |
| Flow of Events | 1. The actor logs into the system. 2. The actor navigates to the data visualization section. 3. The actor chooses the 'Compare Visualized Data' option. 4. The system presents the actor with a selection of datasets or data points for comparison. 5. The actor selects the desired datasets or data points for comparison. 6. The system prompts the actor to confirm the comparison. 7. The actor confirms the comparison. 8. The system performs the comparison. 9. The system notifies the actor. |
| Alternative Flow | N/A |
| Exception Flow - **@Step 5:** The actor chooses datasets or data points that cannot be compared. | The system displays an error message. |
| Exception Flow -  **@Step 8:** Comparison operation fails due to system issues. | The system displays an error message. |

Table 4.2.5: UC - 5 compare visualized data

### 

### **UC - 6 delete data to be visualized**

| Use Case Name | delete data to be visualized |
| --- | --- |
| Use Case ID | UC-6 |
| Description | The system shall allow actors to remove specific datasets or data points from their visualizations. |
| Priority | High |
| Actor(s) | Customer |
| Triggering Event | The actor chooses the 'Delete Data' option within the visualization system. |
| Pre-condition | 1. The actor has valid access to the system. 2. The actor has the necessary permissions to delete data from visualizations. 3. The datasets or data points to be deleted are available and accessible within the visualization. |
| Post-condition | 1. The selected datasets or data points have been deleted from the visualization as per the actor's instructions. 2. The actor is informed of the successful data deletion or of any issues that occurred. |
| Flow of Events | 1. The actor logs into the system. 2. The actor navigates to the data visualization section. 3. The actor chooses the 'Delete Data' option. 4. The system presents the actor with a selection of datasets or data points for deletion. 5. The actor selects the desired datasets or data points for deletion. 6. The system prompts the actor to confirm the deletion. 7. The actor confirms the deletion. 8. The system deletes the selected data. 9. The system notifies the actor. |
| Alternative Flow | N/A |
| Exception Flow -  **@Step 5:** The actor chooses datasets or data points that cannot be deleted. | The system displays an error message. |
| Exception Flow -  **@Step 8:** Deletion operation fails due to system issues. | The system displays an error message. |

Table 4.2.6: UC - 6 delete data to be visualized

### 

### **UC - 7 update data to be visualized**

| Use Case Name | update data to be visualized |
| --- | --- |
| Use Case ID | UC - 7 |
| Description | The system shall allow the actor to add new data to their existing visualizations. |
| Priority | High |
| Actor(s) | Customer |
| Triggering Event | The actor clicks the “update” button. |
| Pre-condition | The actor has created a visualization of the data. |
| Post-condition | The updated data is reflected in the visualization. |
| Flow of Events | 1. The actor selects the visualization they want to update. 2. The actor clicks the “update” button. 3. The system retrieves the current data associated with the selected visualization. 4. The actor adds new data. 5. The system validates the updated data to ensure it conforms to the required format and structure. 6. The system updates the visualization with the new data. 7. The actor views the updated visualization with the modified or added data. |
| Alternative Flow - @**Step 2:** The actor wants to discard the changes. | 1. The actor clicks the “cancel” button. 2. Back to @Step 1. |
| Exception Flow - @**Step 5:** The actor inputs invalid data. | The system displays an error message. |

Table 4.2.7: UC - 7 update data to be visualized

### **UC - 8 zoom in and out data visualization**

| Use Case Name | zoom in and out data visualization |
| --- | --- |
| Use Case ID | UC - 8 |
| Description | The system shall allow the actor to magnify or reduce their view of a data visualization, enabling them to see details or the big picture as needed. |
| Priority | Moderate |
| Actor(s) | Customer |
| Triggering Event | The actor clicks the “zoom in” or “zoom out” icon of the visualization. |
| Pre-condition | The actor has created a visualization of the data. |
| Post-condition | The data visualization is displayed at the desired zoom level. |
| Flow of Events | 1. The actor selects the data visualization they want to zoom in or out. 2. The system displays the selected visualization. 3. The actor clicks the “zoom in” or “zoom out” icon of the visualization. 4. The system adjusts the zoom level of the visualization accordingly. 5. The system re-renders the visualization to reflect the new zoom level. 6. The actor views the data visualization at the adjusted zoom level. |
| Alternative Flow -  @**Step 6:** The actor wants to reset the zoom level to the default state. | 1. The actor selects a "Reset Zoom" option provided by the system. 2. Back to @Step 2. |
| Exception Flow | N/A |

Table 4.2.8: UC - 8 zoom in and out data visualization

### **UC - 9 filter and sort data**

| Use Case Name | filter and sort data |
| --- | --- |
| Use Case ID | UC - 9 |
| Description | The system shall allow the actor to filter or sort data in a specific order before or after visualization. |
| Priority | High |
| Actor(s) | Customer |
| Triggering Event | The actor clicks the “filter” or “sort” icon of the visualization. |
| Pre-condition | The actor has created a visualization of the data. |
| Post-condition | The data is filtered or sorted according to the actor’s specifications for visualization. |
| Flow of Events | 1. The actor selects the data visualization they want to apply filters or sorting to. 2. The system displays the selected visualization. 3. The actor indicates their desire to apply filters or sorting. 4. The system presents options to define filters and sorting. 5. The actor specifies the desired filter conditions (e.g., selecting specific values, defining ranges) and/or sorting criteria (e.g., ascending, descending order). 6. The system validates and applies the specified filters and/or sorting to the data. 7. The system updates the visualization to reflect the filtered and/or sorted data. 8. The actor views the data visualization with the filtered and/or sorted data. |
| Alternative Flow | N/A |
| Exception Flow | N/A |

Table 4.2.9: UC - 9 filter and sort data

### **UC - 10 change visualization technique**

| Use Case Name | change visualization technique |
| --- | --- |
| Use Case ID | UC - 10 |
| Description | The system shall allow the actor to conveniently change the method used to visualize their data, enabling them to switch between different visualization techniques. |
| Priority | High |
| Actor(s) | Customer |
| Triggering Event | The actor selects the new visualization technique. |
| Pre-condition | The actor has created a visualization of the data. |
| Post-condition | The data is displayed using the newly selected visualization technique. |
| Flow of Events | 1. The actor selects the data visualization they want to change the technique for. 2. The system displays the selected visualization. 3. The system presents the user with a list of available visualization techniques or options. 4. The actor selects the desired visualization technique from the provided options. 5. The system reconfigures the visualization using the newly selected technique. 6. The system updates the visualization to reflect the changes made. 7. The actor views the data using the new visualization technique. |
| Alternative Flow -  @**Step 7:** The actor wants to revert to the previous visualization technique. | 1. The actor clicks the "Undo" icon. 2. Back to @Step 2. |
| Exception Flow | N/A |

Table 4.2.10: UC - 10 change visualization technique

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### **UC - 11 change data to be visualized**

| Use Case Name | change data to be visualized |
| --- | --- |
| Use Case ID | UC - 11 |
| Description | The system shall enable the actor to select or deselect different datasets or data points for visualization, allowing them to modify the data being presented. |
| Priority | High |
| Actor(s) | Customer |
| Triggering Event | The actor selects or deselects new datasets or data points. |
| Pre-condition | The actor has created a visualization of the data. |
| Post-condition | The visualization reflects the newly selected datasets or data points. |
| Flow of Events | 1. The actor selects the data visualization set they want to modify. 2. The system displays the selected visualization. 3. The actor indicates their desire to change the data being visualized. 4. The system presents the user with options to select or deselect datasets or data points. 5. The actor selects or deselects the desired datasets or data points for visualization. 6. The system updates the visualization to reflect the newly selected data. 7. The actor views the visualization with the modified data. |
| Alternative Flow | N/A |
| Exception Flow -  @**Step 6:** The selected datasets or data points are not available or do not match the visualization's requirements. | The system displays an error message and provides alternative options or suggestions. |

Table 4.2.11: UC - 11 change data to be visualized

### 

### **UC - 12 change data visualization colour**

| Use Case Name | change data visualization colour |
| --- | --- |
| Use Case ID | UC - 12 |
| Description | The system shall allow the actor to customize the color scheme of their data visualization for better readability or aesthetics. |
| Priority | Moderate |
| Actor(s) | Customer |
| Triggering Event | The actor selects the new color scheme of the data visualization. |
| Pre-condition | The actor has created a visualization of the data. |
| Post-condition | The data visualization is displayed using the newly selected color scheme. |
| Flow of Events | 1. The actor selects the data visualization they want to modify. 2. The system displays the selected visualization. 3. The actor indicates their desire to change the color scheme of the visualization. 4. The system presents the actor with options to customize the color scheme. 5. The actor selects the desired colors for different elements of the visualization (e.g., data points, background, labels). 6. The system updates the visualization with the newly selected color scheme. 7. The actor views the data visualization with the modified color scheme. |
| Alternative Flow -  @**Step 7:** The actor wants to revert to the default color scheme. | The actor selects the "Undo" icon. |
| Exception Flow | N/A |

Table 4.2.12: UC - 12 change data visualization colour

### 

### **UC - 13 view visualized data in iOS and Android apps**

| Use Case Name | view visualized data in iOS and Android apps |
| --- | --- |
| Use Case ID | UC - 13 |
| Description | The system shall enable the actor to access and view visualized data on both iOS and Android platforms through dedicated mobile applications. |
| Priority | High |
| Actor(s) | Customer |
| Triggering Event | The actor opens the mobile application. |
| Pre-condition | The actor has installed the mobile application on their iOS or Android device. |
| Post-condition | The actor can view visualized data on their iOS or Android device. |
| Flow of Events | 1. The actor launches the iOS or Android app on their device. 2. The system displays the login screen or user dashboard. 3. The actor enters their login credentials and authenticates themselves. 4. The system verifies the actor’s credentials and grants access to the user dashboard. 5. The actor selects the desired visualized data from their dashboard. 6. The system retrieves the selected visualized data from the server. 7. The system renders and displays the visualized data on the actor 's iOS or Android device. 8. The actor views and interacts with the visualized data on their device. |
| Alternative Flow | N/A |
| Exception Flow - @**Step 4:** The actor encounters issues during the login/authentication process. | The system displays error messages. |

Table 4.2.13: UC - 13 view visualized data in iOS and Android apps

## 

## **4.3 UML Class Diagram**

## **4.4 UML Sequence Diagram**

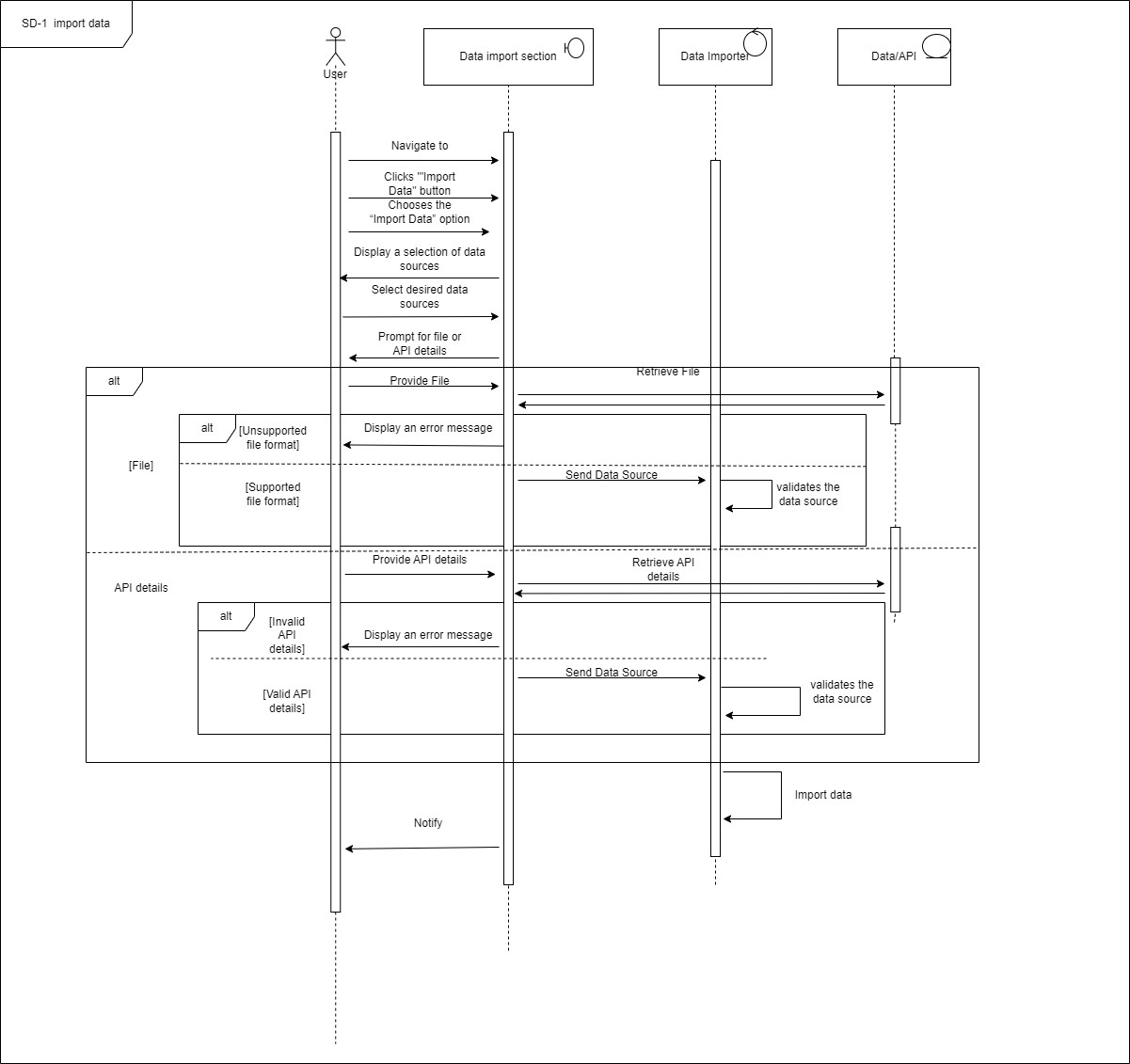


Figure 4.4.1: Sequence Diagram SD - 1 import data

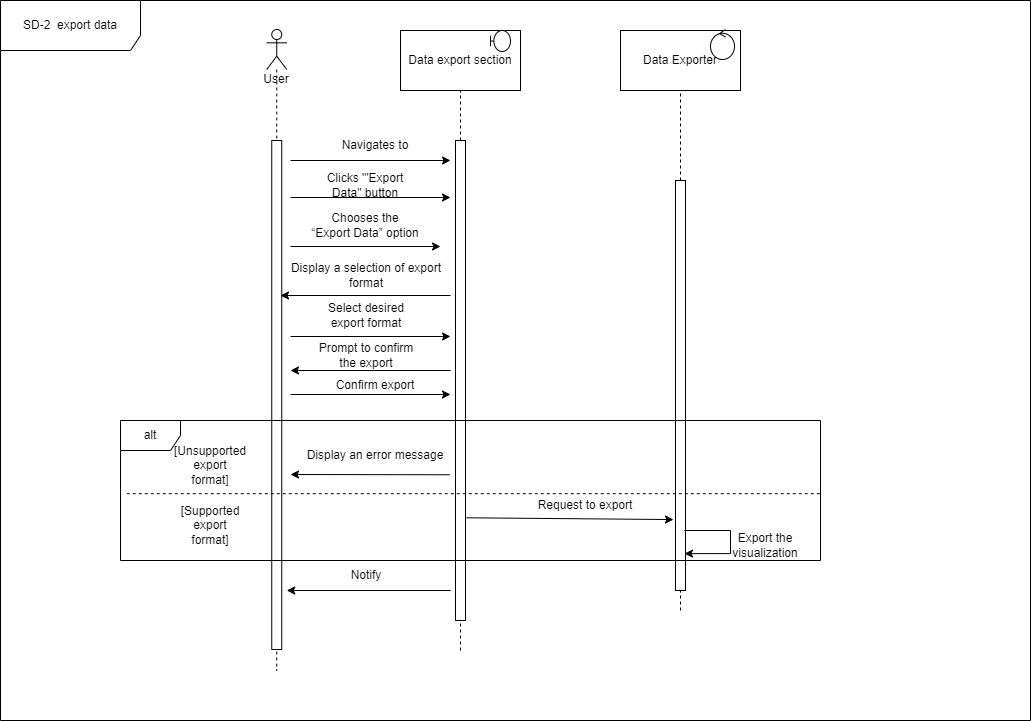


Figure 4.4.2: Sequence Diagram SD - 2 export data

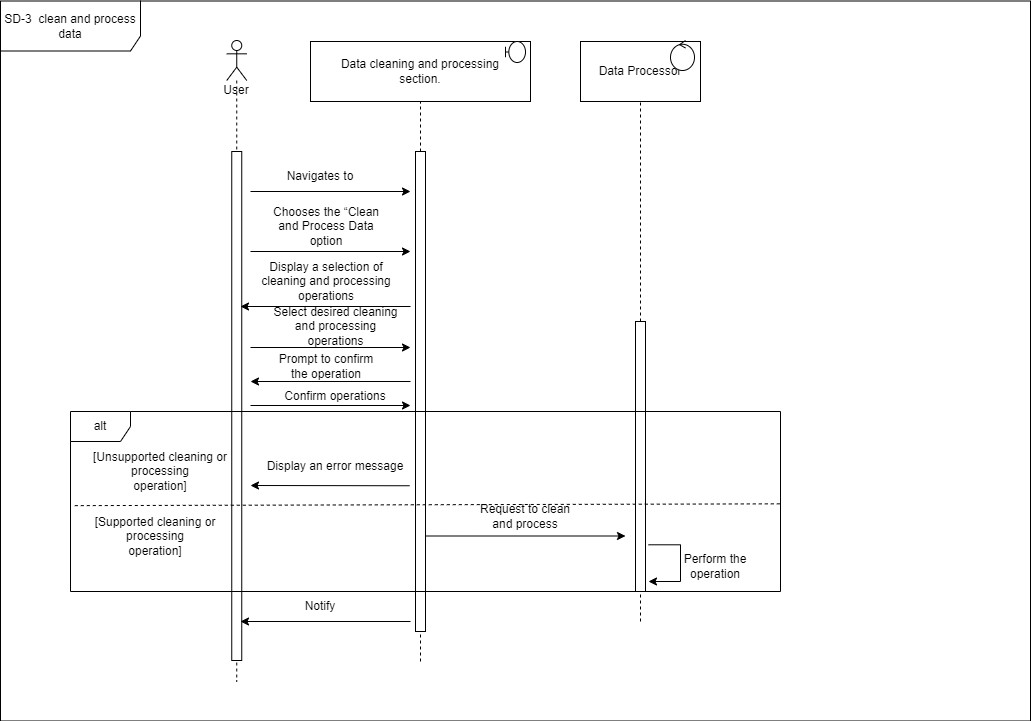


Figure 4.4.3: Sequence Diagram SD - 3 clean and process data

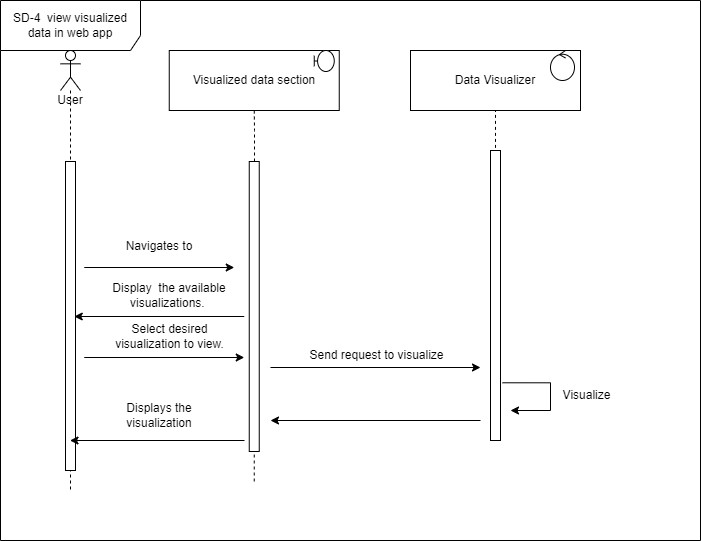


Figure 4.4.4: Sequence Diagram SD - 4 view visualized data in web app

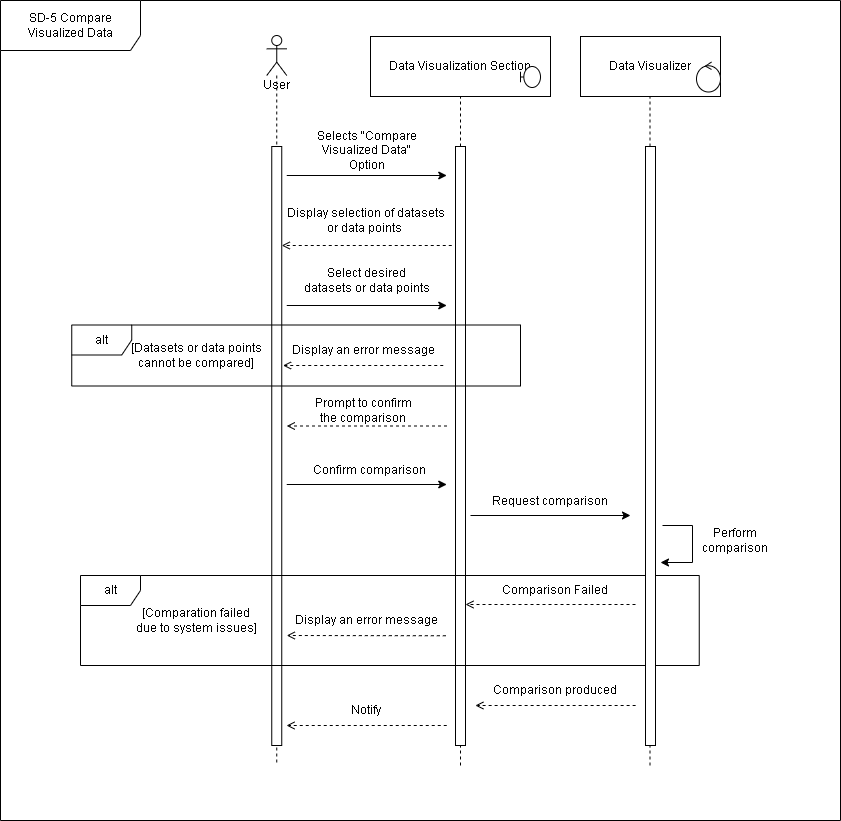


Figure 4.4.5: Sequence Diagram SD - 5 compare visualized data

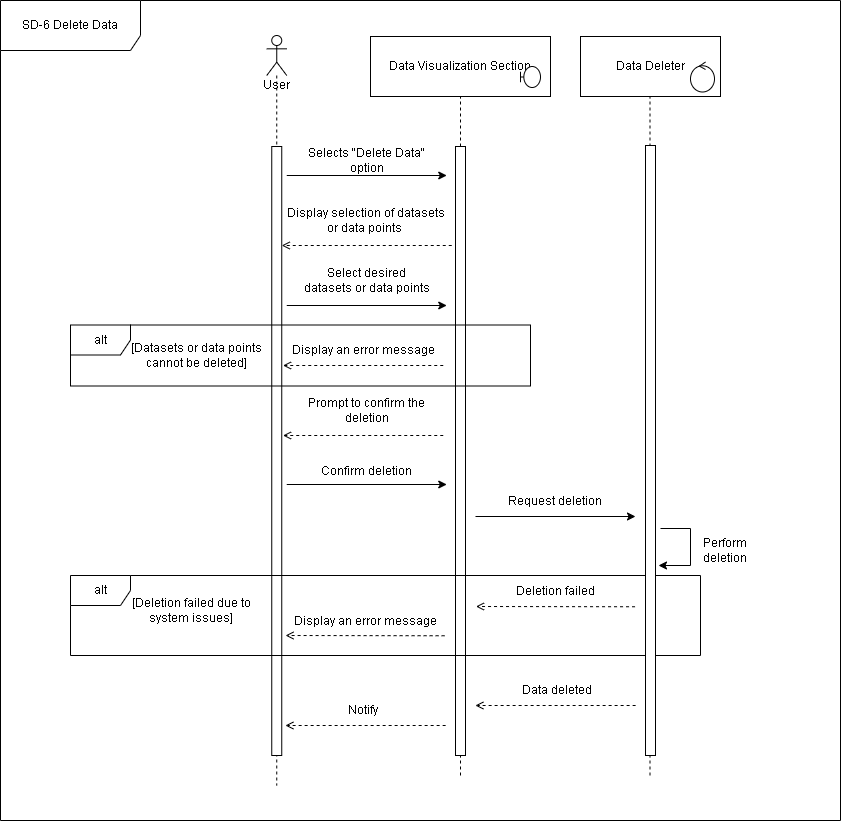


Figure 4.4.6: Sequence Diagram SD - 6 delete data to be visualized

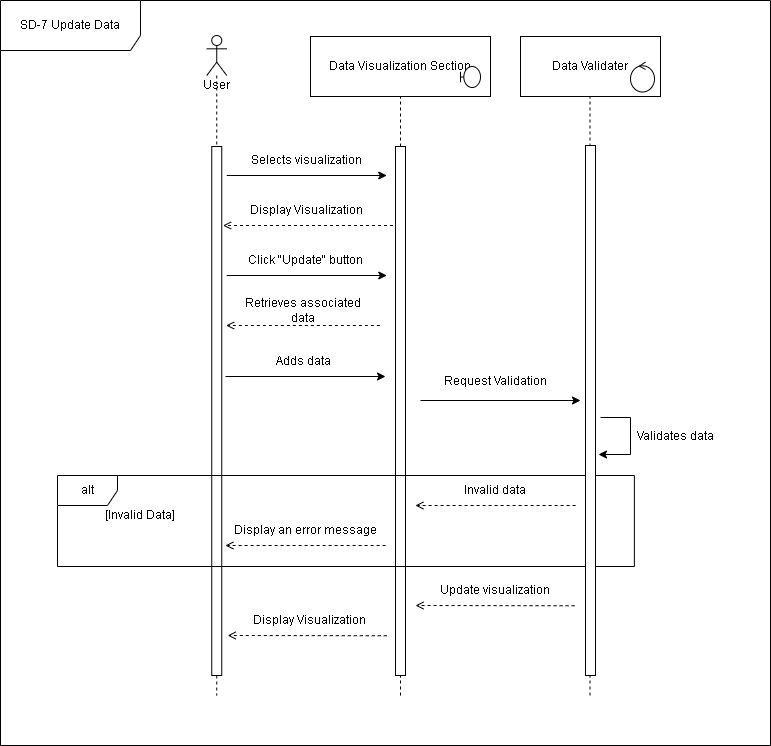


Figure 4.4.7: Sequence Diagram SD - 7 update data to be visualized

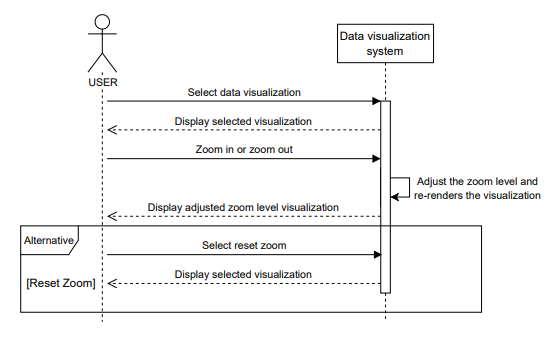


Figure 4.4.8: Sequence Diagram SD - 8 zoom in and zoom out data visualization

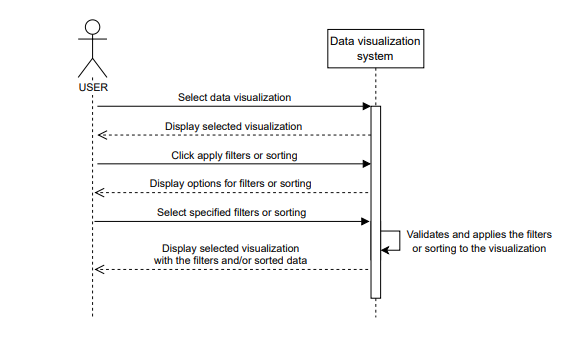


Figure 4.4.9: Sequence Diagram SD - 9 filter and sort data visualization

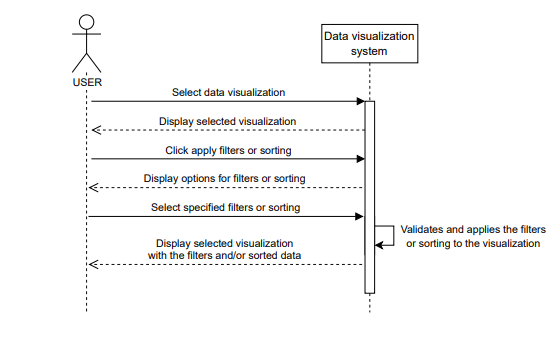


Figure 4.4.10: Sequence Diagram SD - 10 change visualization technique

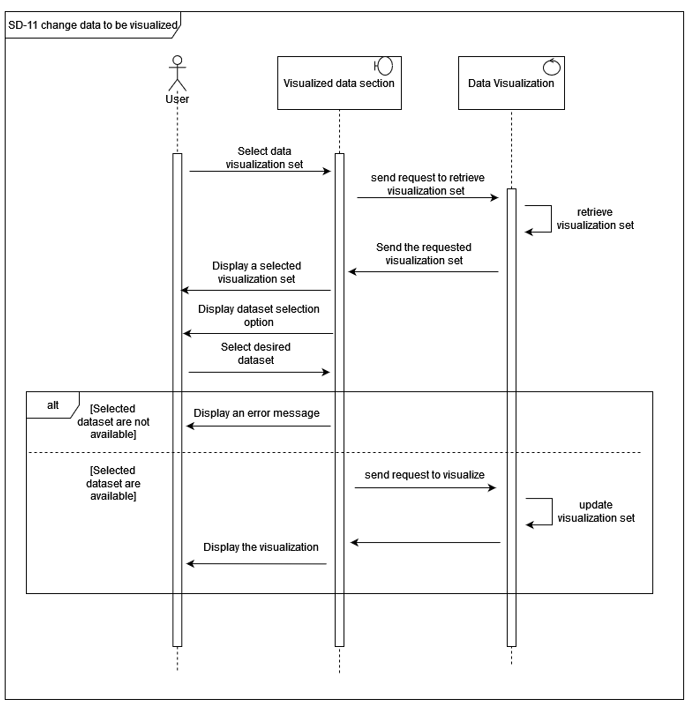


Figure 4.4.11: Sequence Diagram SD - 11 change data to be visualized

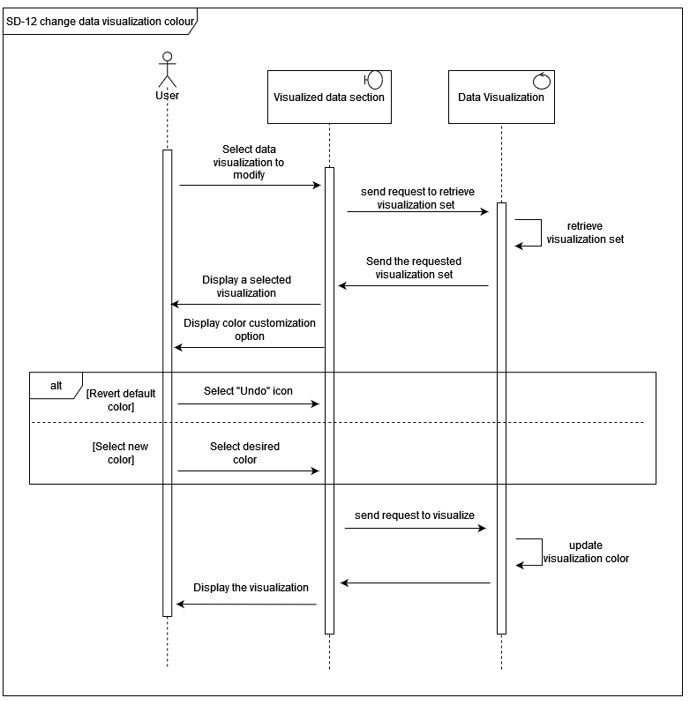


Figure 4.4.12: Sequence Diagram SD - 12 change data visualization colour

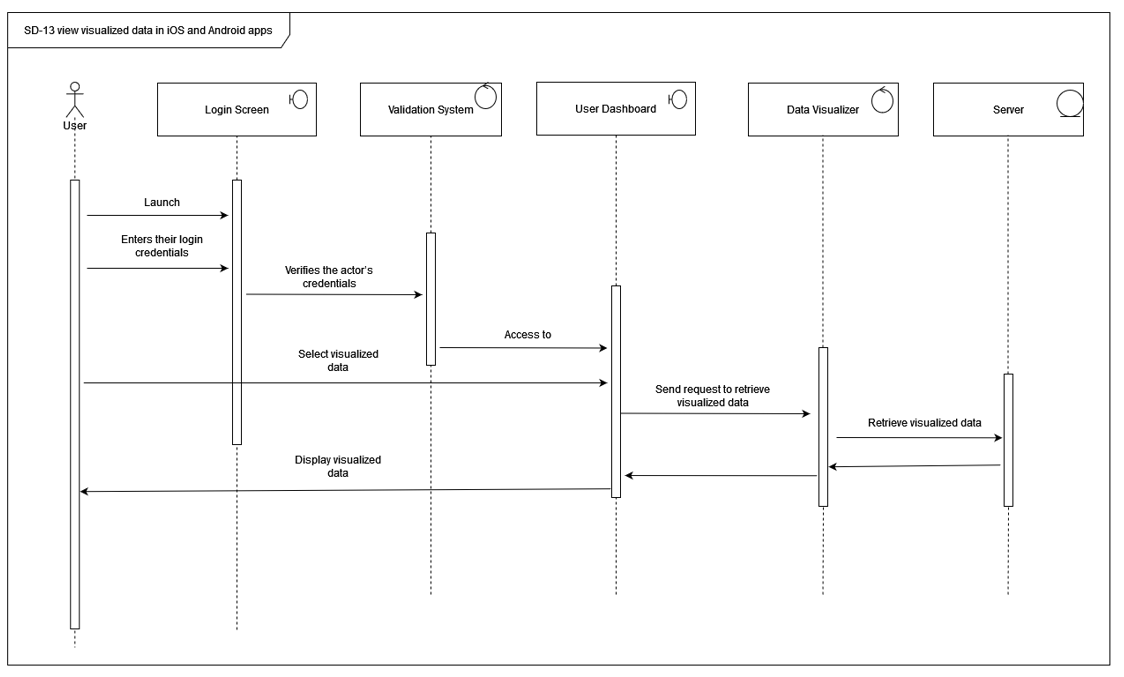


Figure 4.4.13: Sequence Diagram SD - 13 view visualized data in iOS and Android apps

## **4.5 UML Activity Diagram**

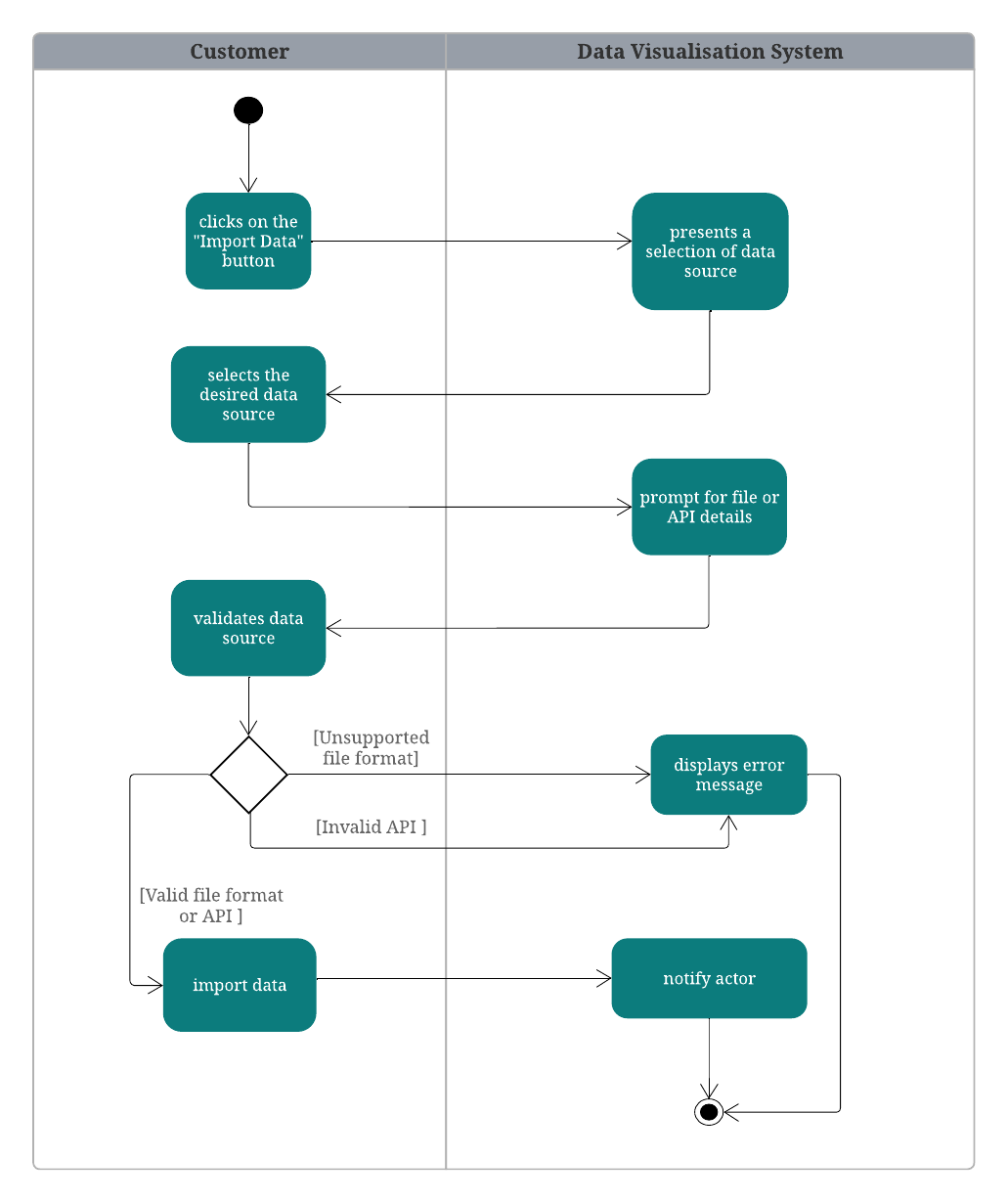


Figure 4.5.1: Activity Diagram UC - 1 import data

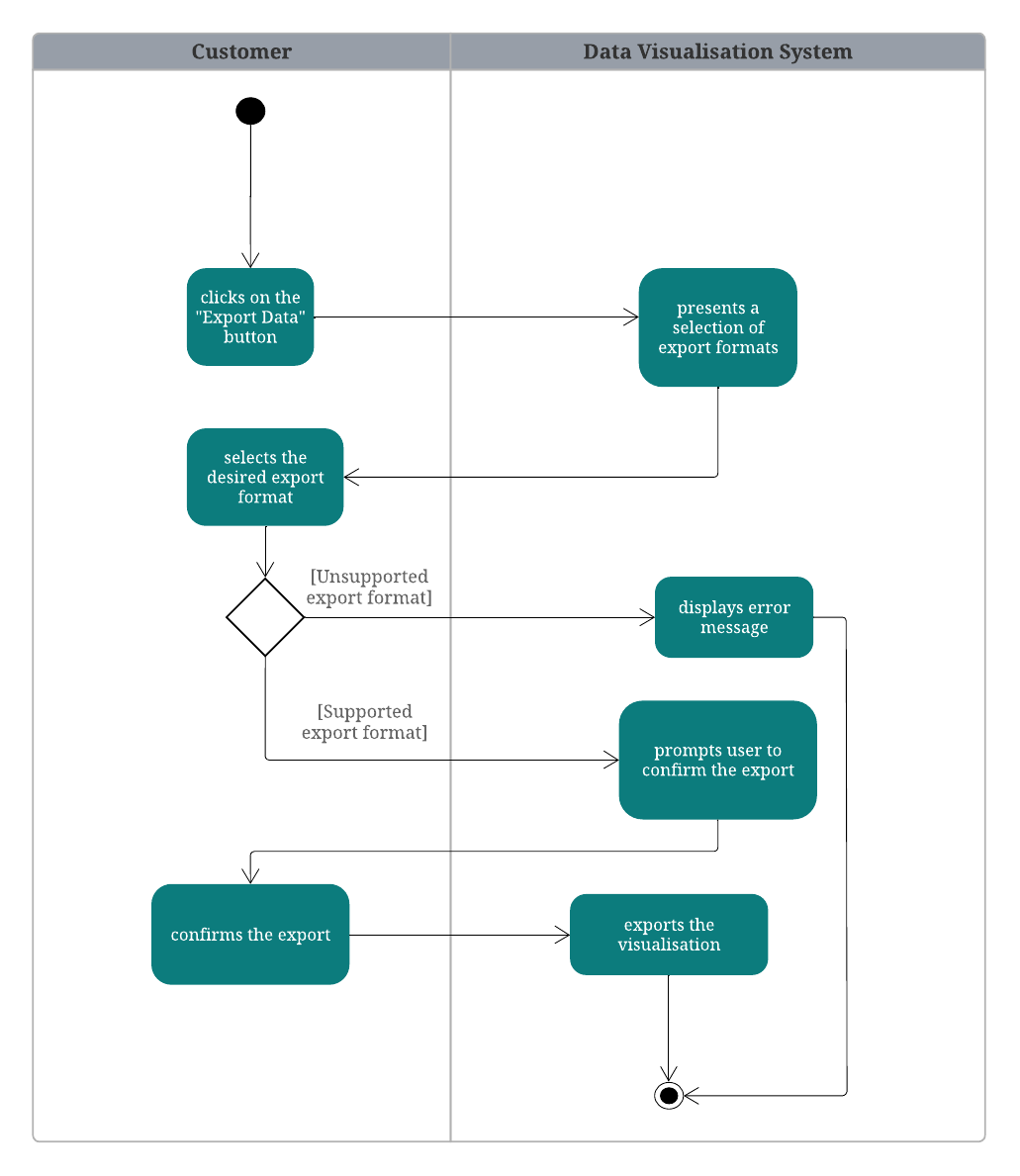


Figure 4.5.2: Activity Diagram UC - 2 export data

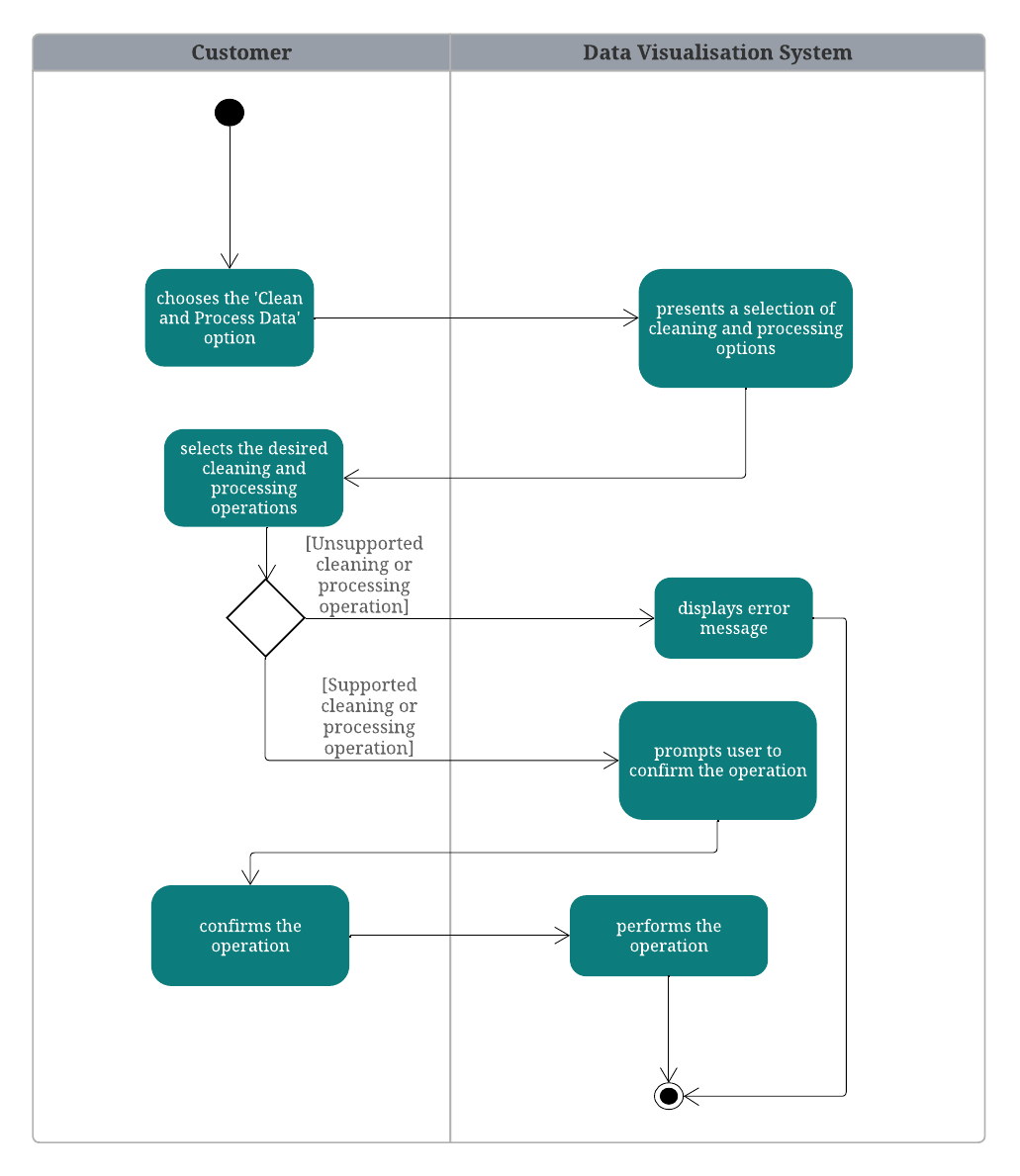


Figure 4.5.3: Activity Diagram UC - 3 clean and process data

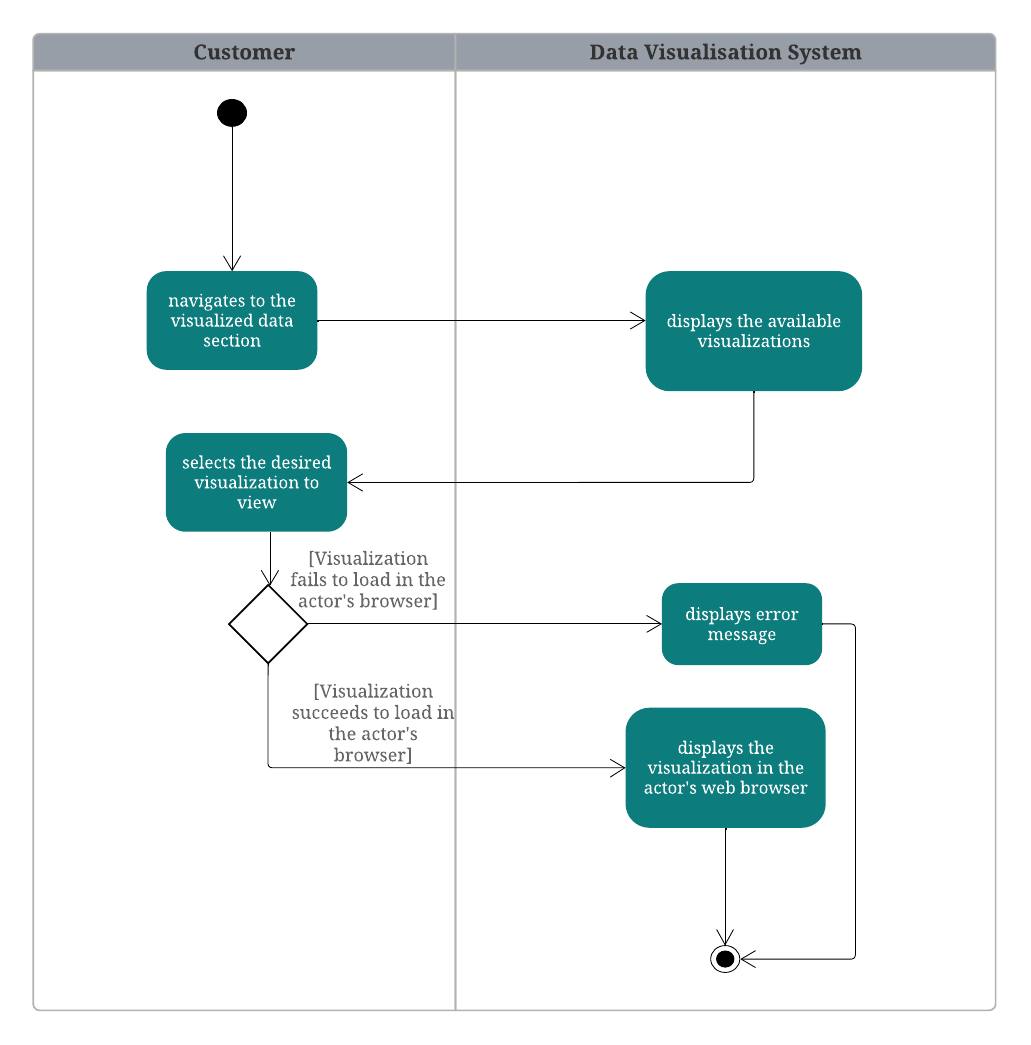


Figure 4.5.4: Activity Diagram UC - 4 view visualized data in web app

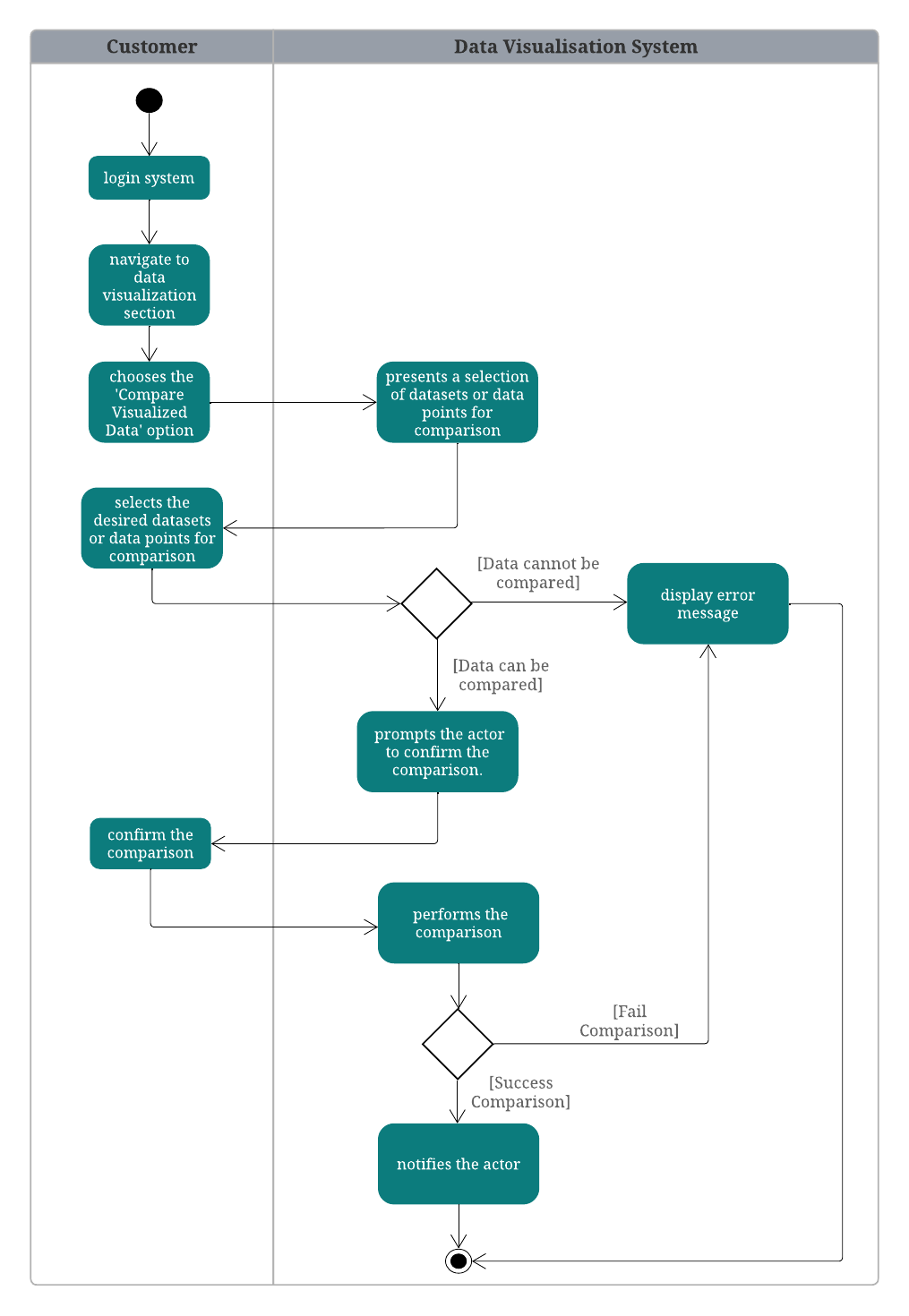


Figure 4.5.5: Activity Diagram UC - 5 compare visualized data

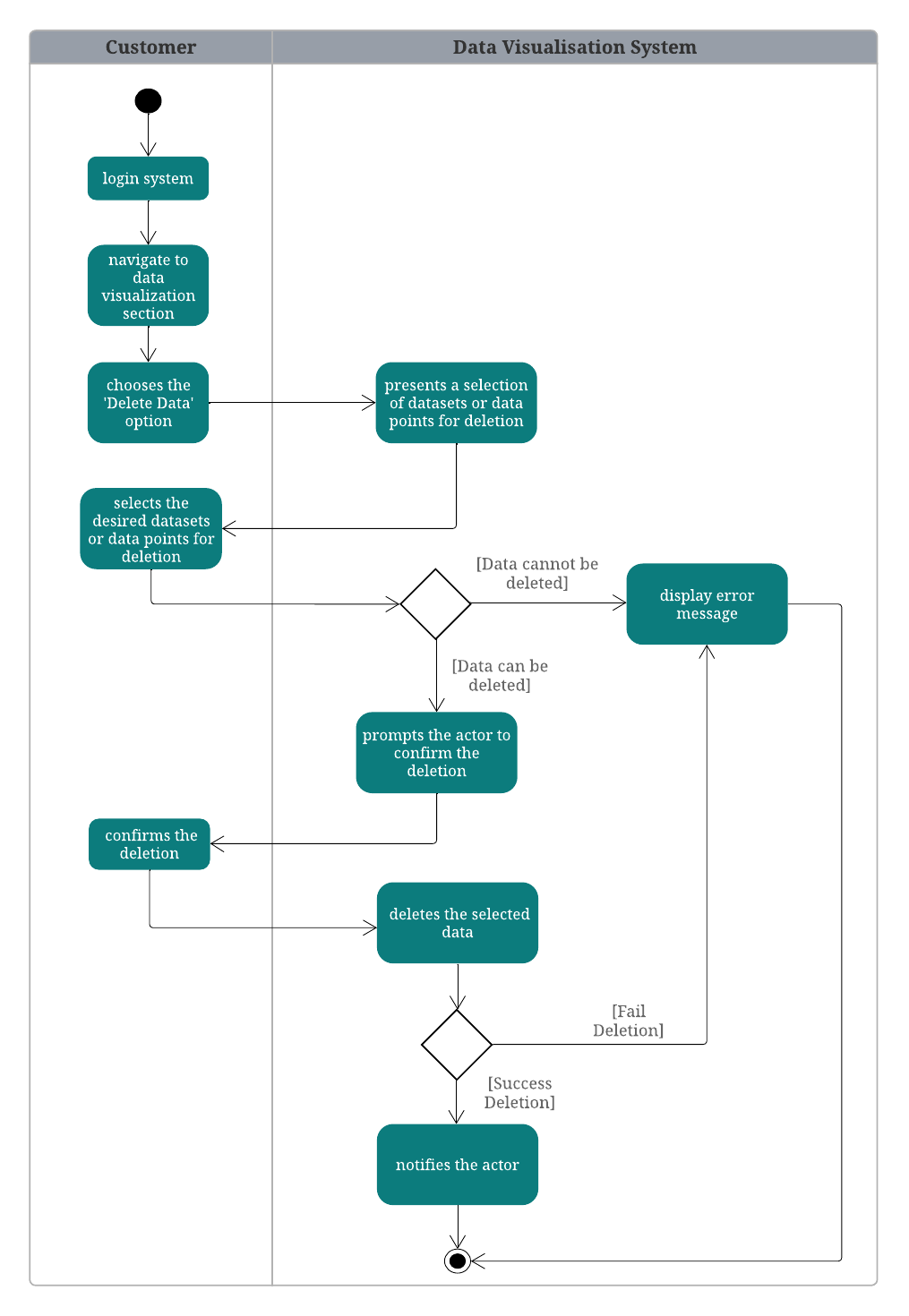


Figure 4.5.6: Activity Diagram UC - 6 delete data to be visualized

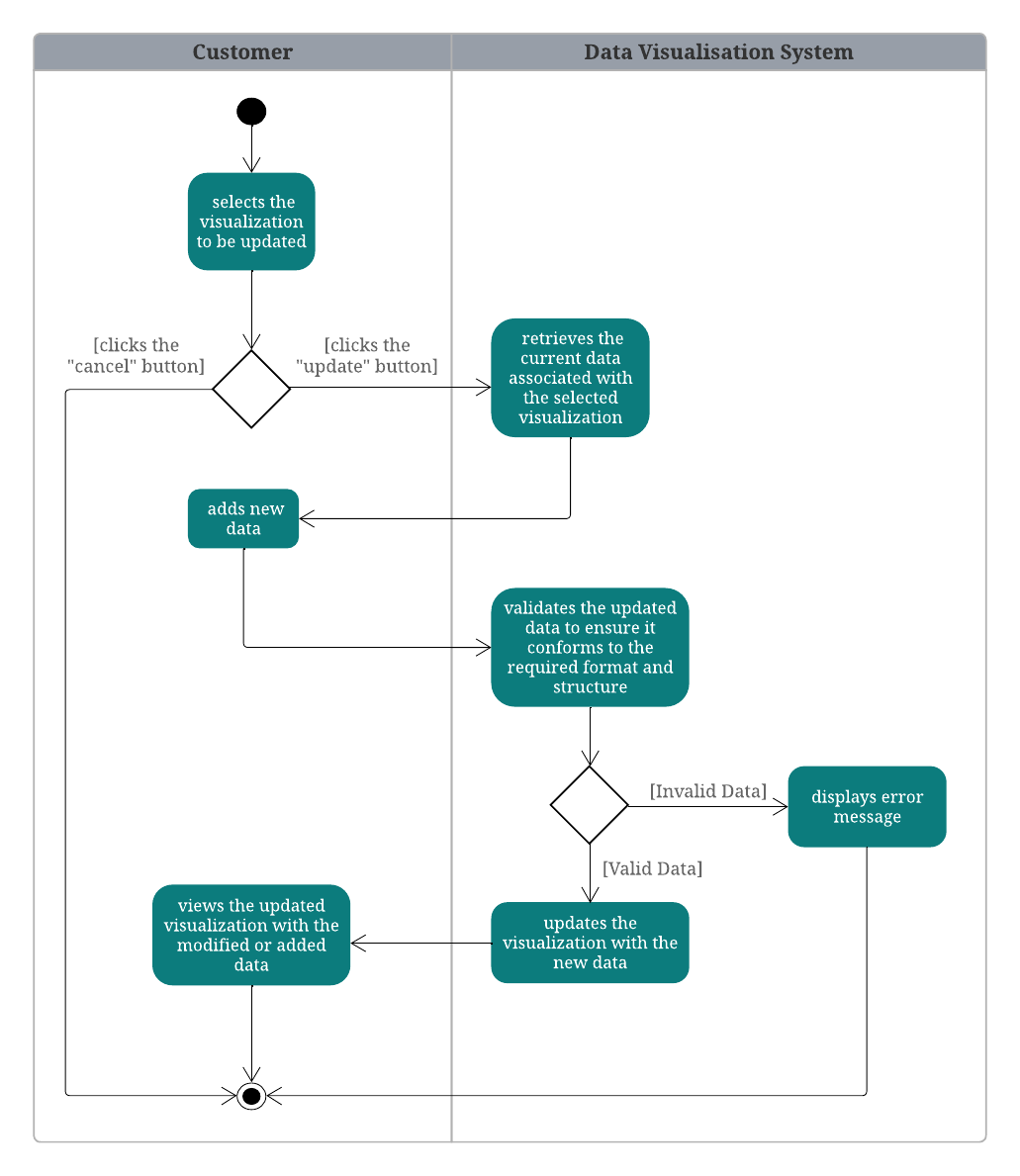


Figure 4.5.7: Activity Diagram UC - 7 update data to be visualized

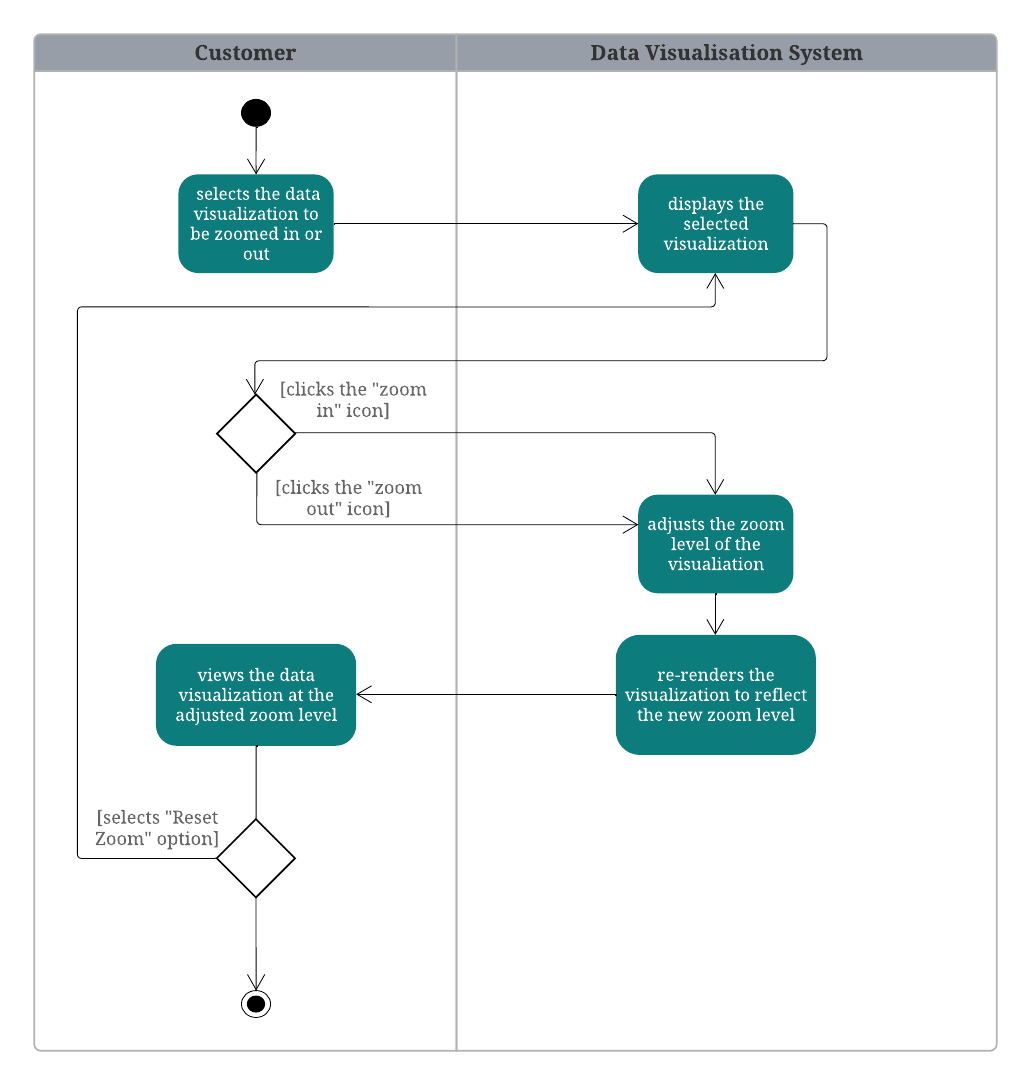


Figure 4.5.8: Activity Diagram UC - 8 zoom in and out data visualization

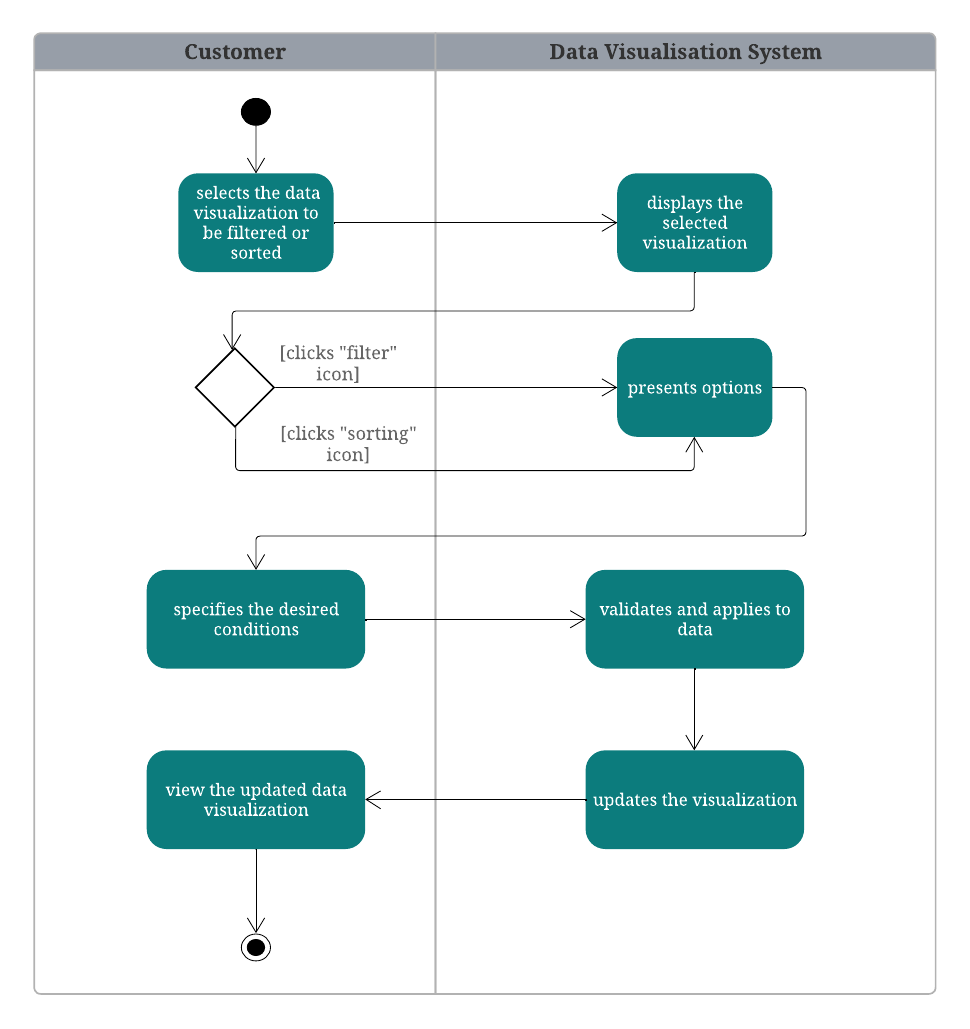


Figure 4.5.9: Activity Diagram UC - 9 filter and sort data

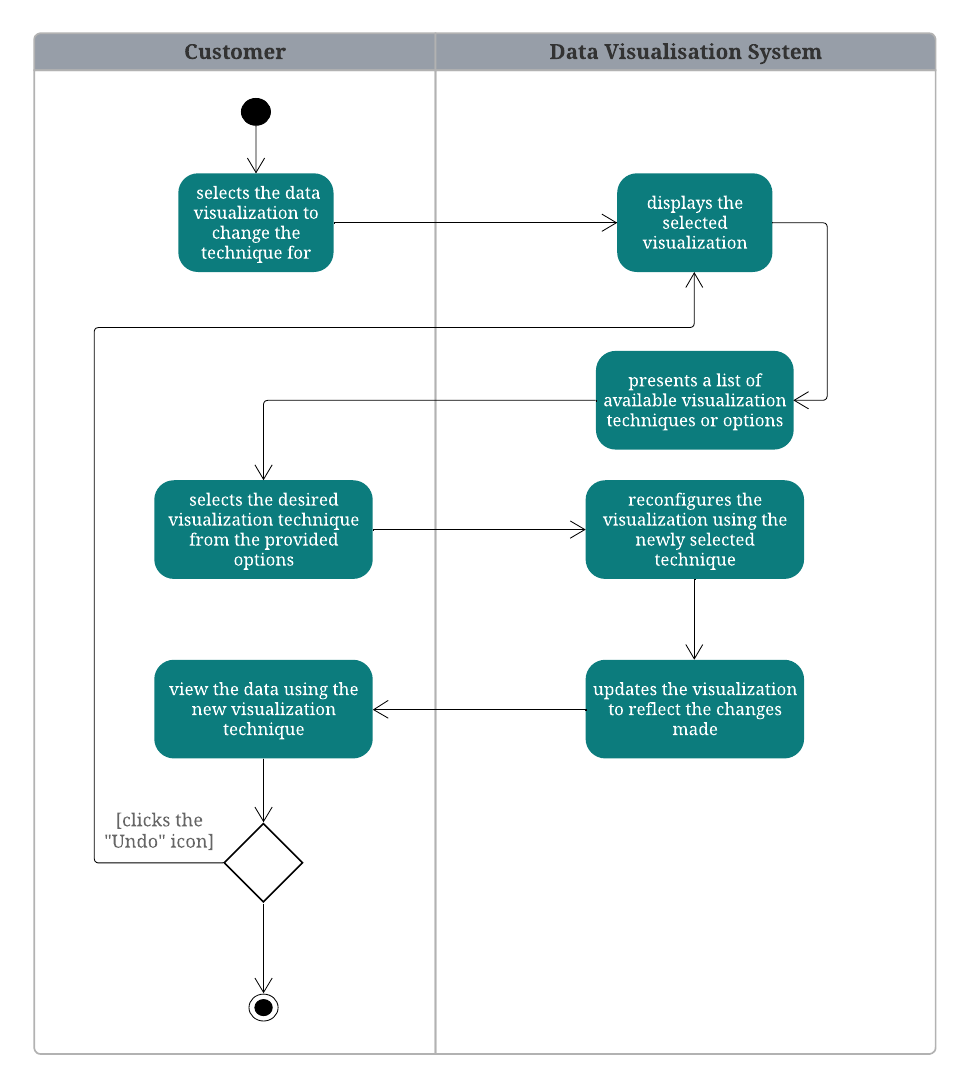


Figure 4.5.10: Activity Diagram UC - 10 change visualization technique

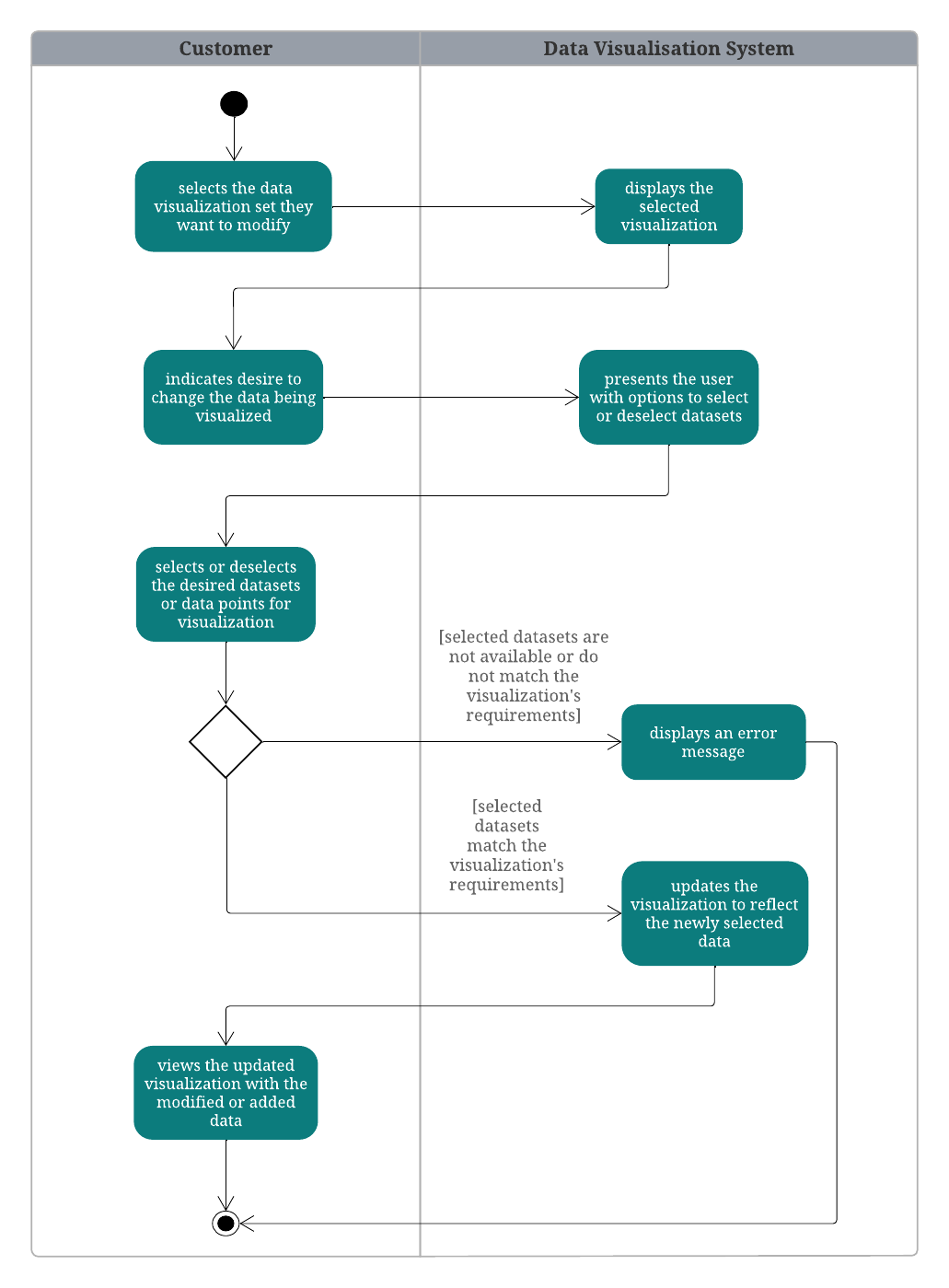


Figure 4.5.11: Activity Diagram UC - 11 change data to be visualized

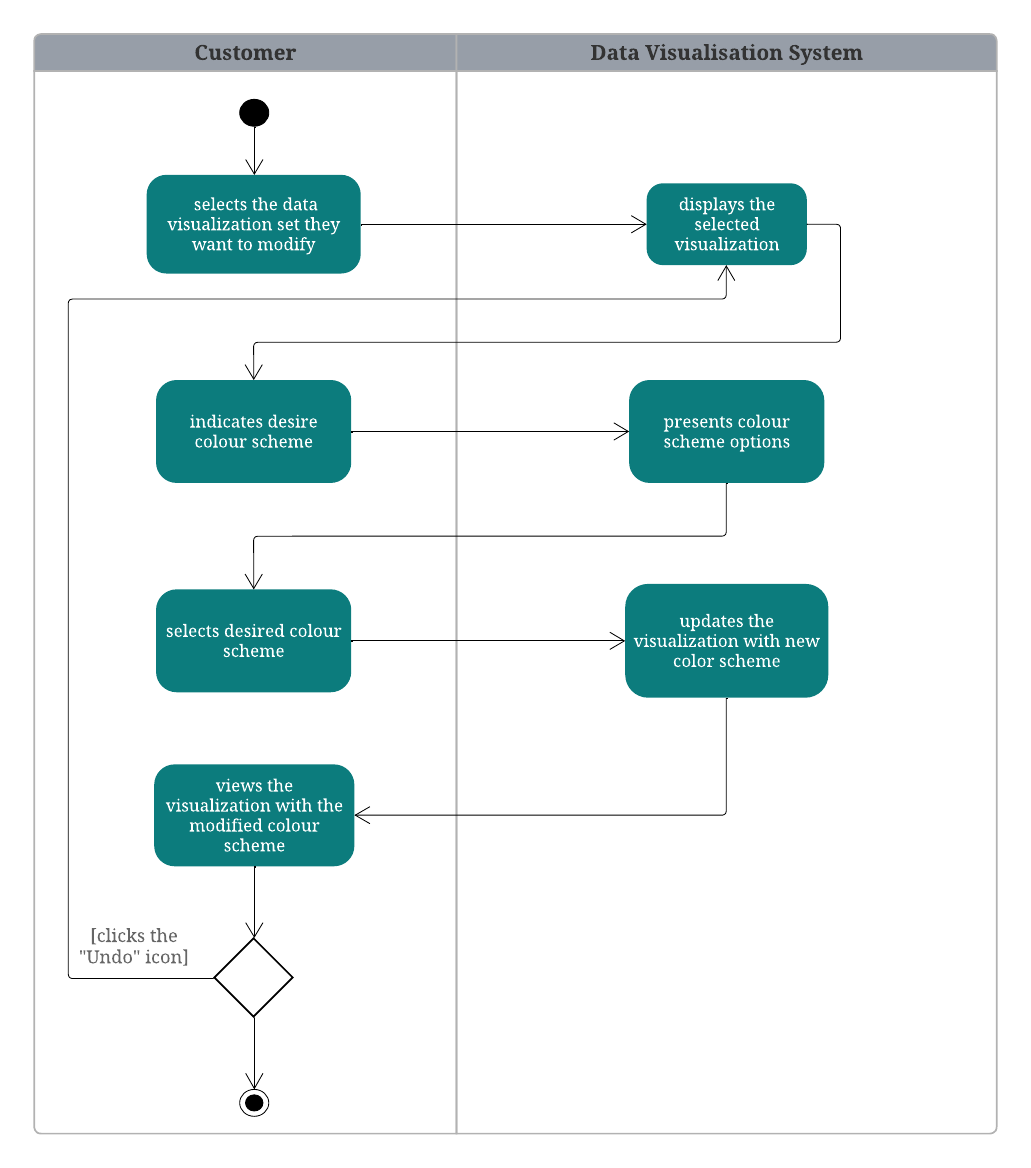


Figure 4.5.12: Activity Diagram UC - 12 change data visualization colour

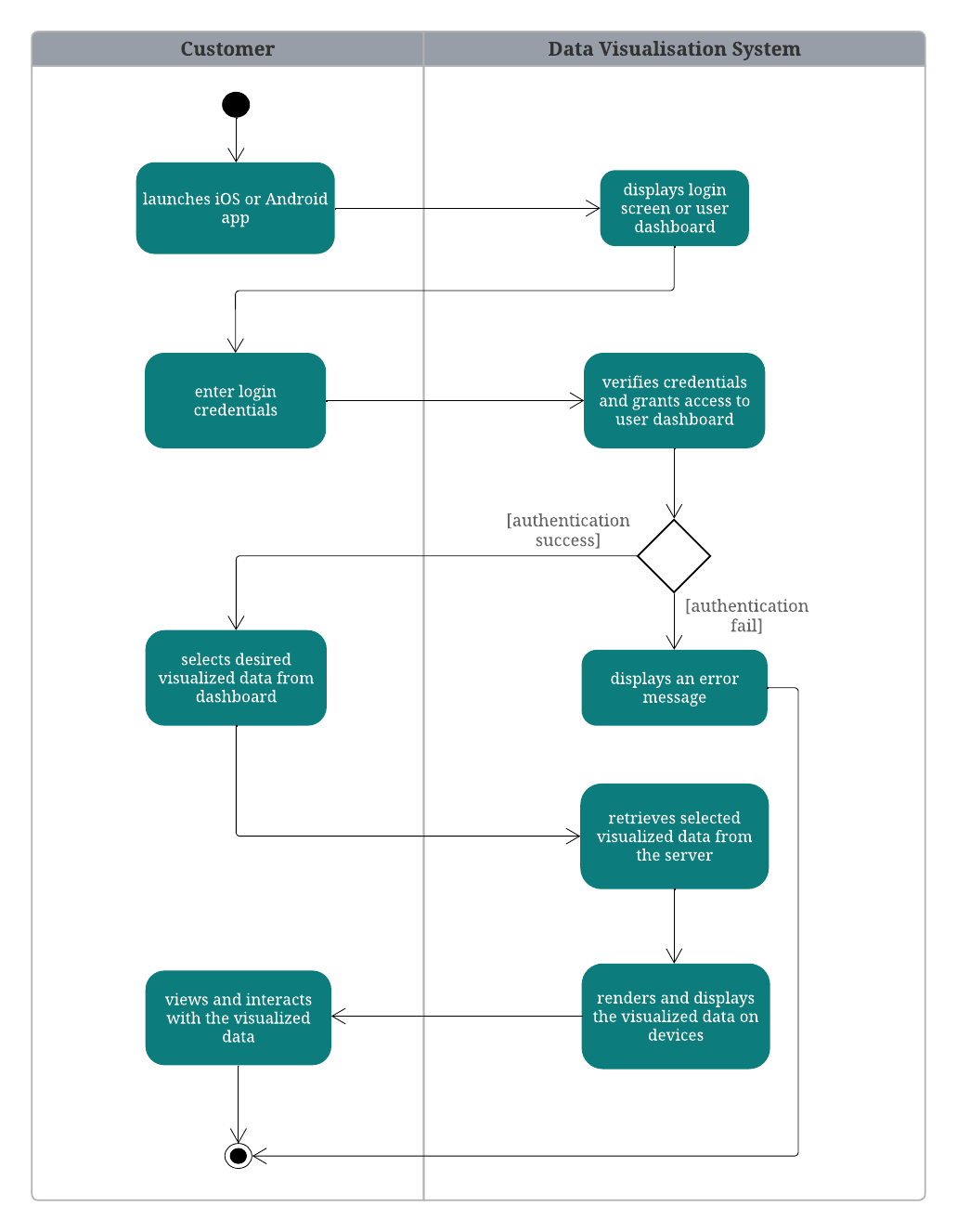


Figure 4.5.13: Activity Diagram UC - 13 view visualized data in iOS and Android apps

## **4.6 UML Component Diagram**

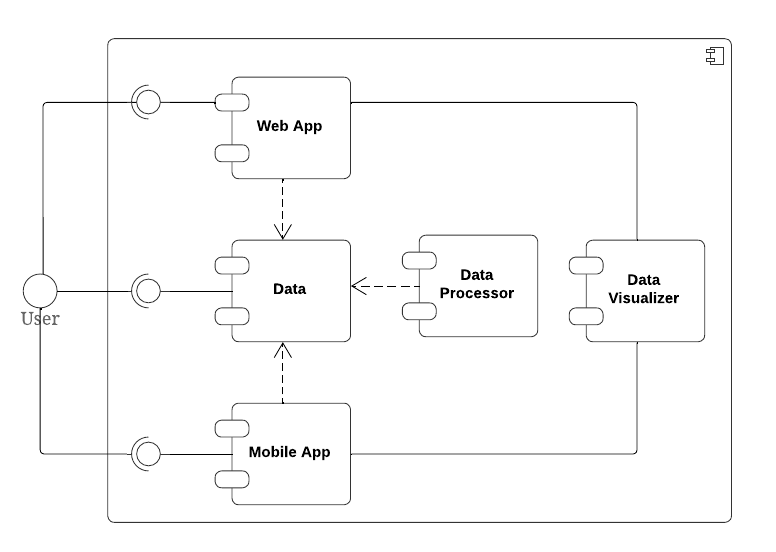


Figure 4.6.1: Component Diagram

## 

## **4.7 UML Deployment Diagram**

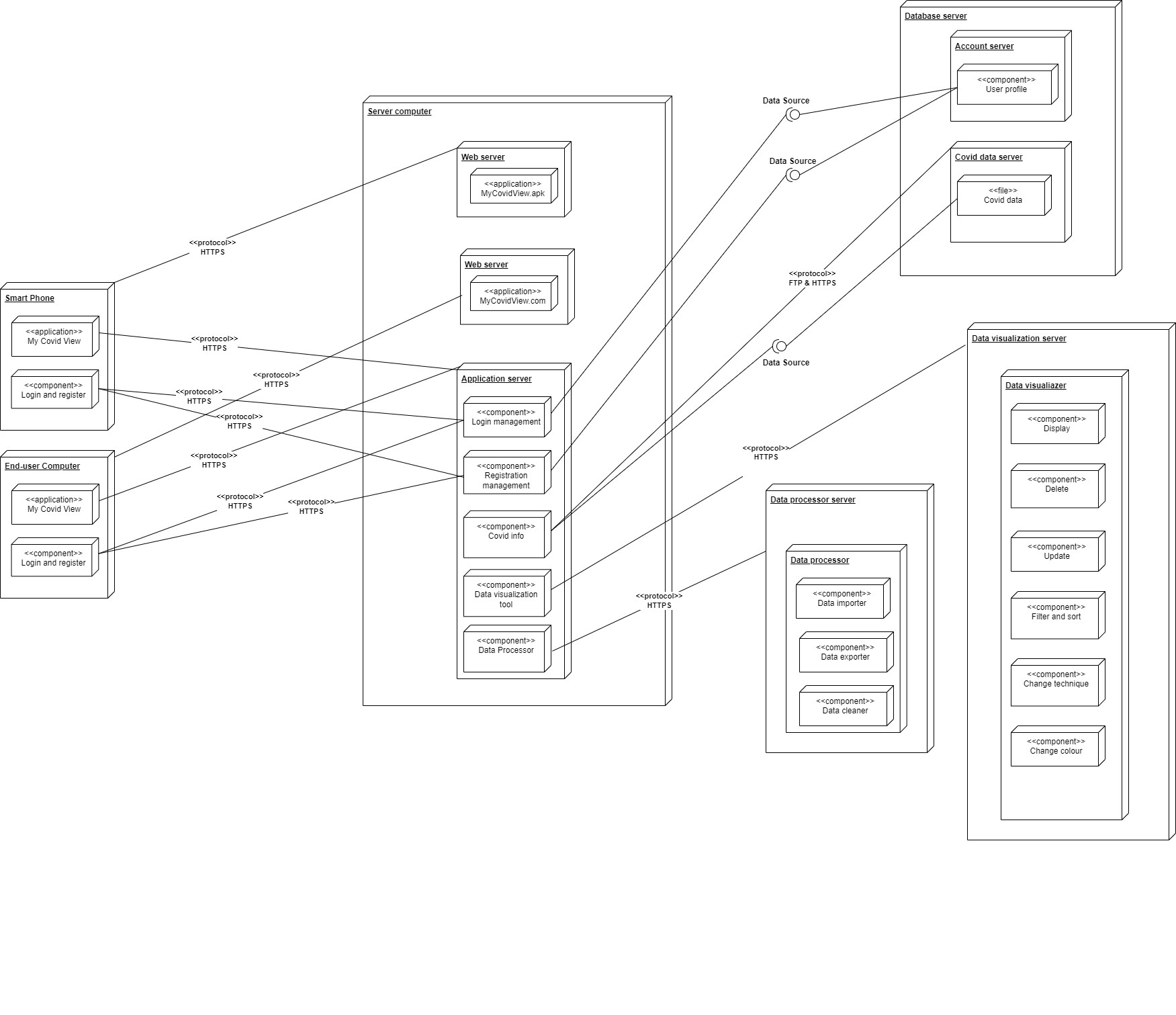


Figure 4.7.1: Deployment Diagram

# 

# USER INTERFACE REQUIREMENTS

## **5.1 Mockup screen (Wireframe Version 1)**

### **5.1.1 Web View(Monitor 1920px wide)**

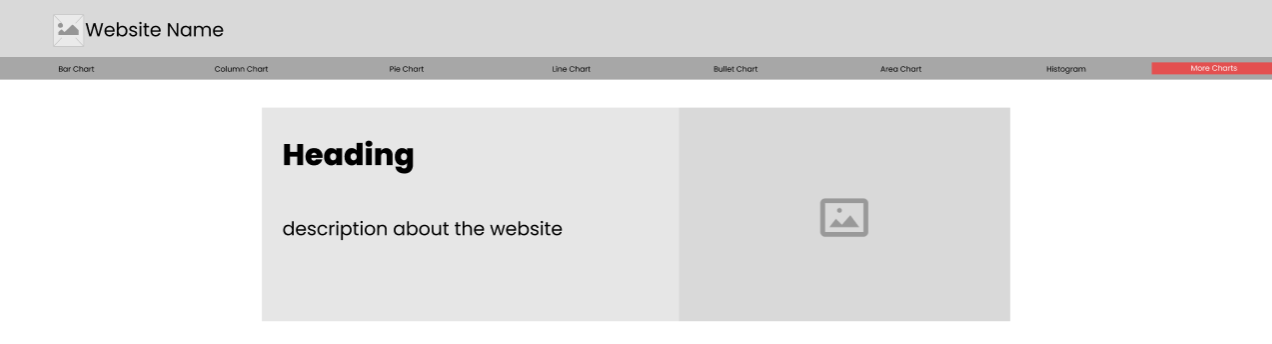


Figure 5.1.1.1 The main page of the application

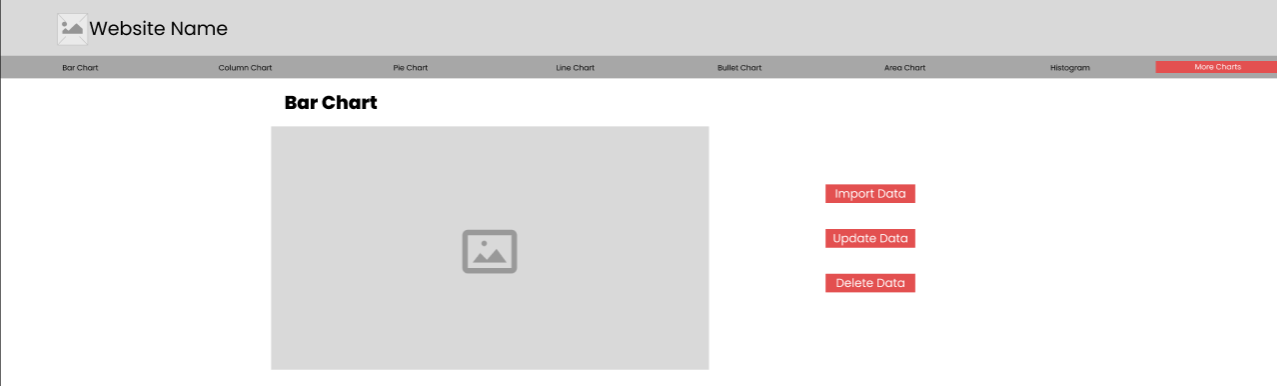
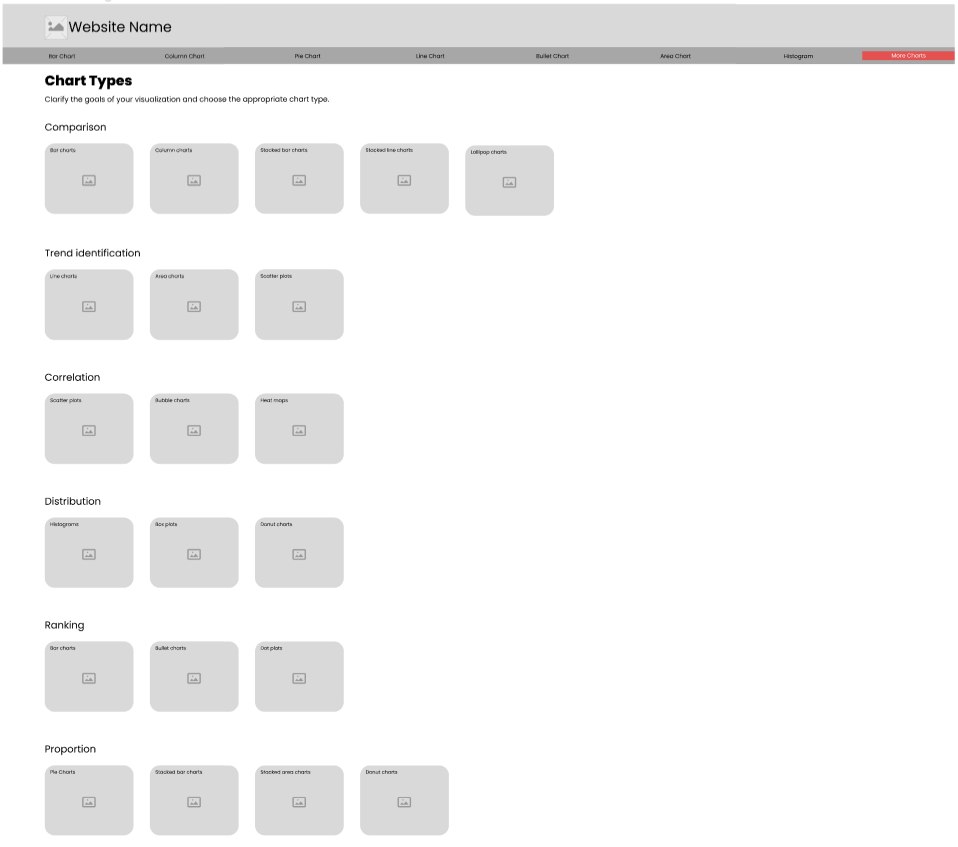


Figure 5.1.1.2 The main page of a chart



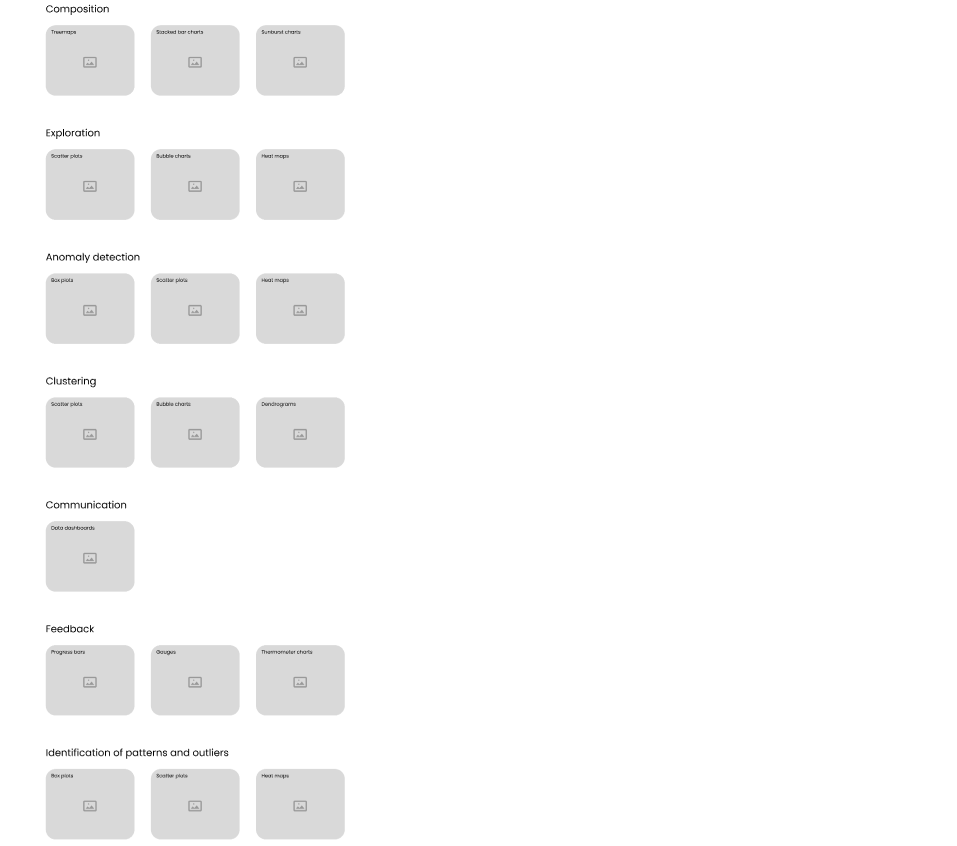




Figure 5.1.1.3 The chart type page

### **5.1.2 Web View(Laptop 1024px wide)**

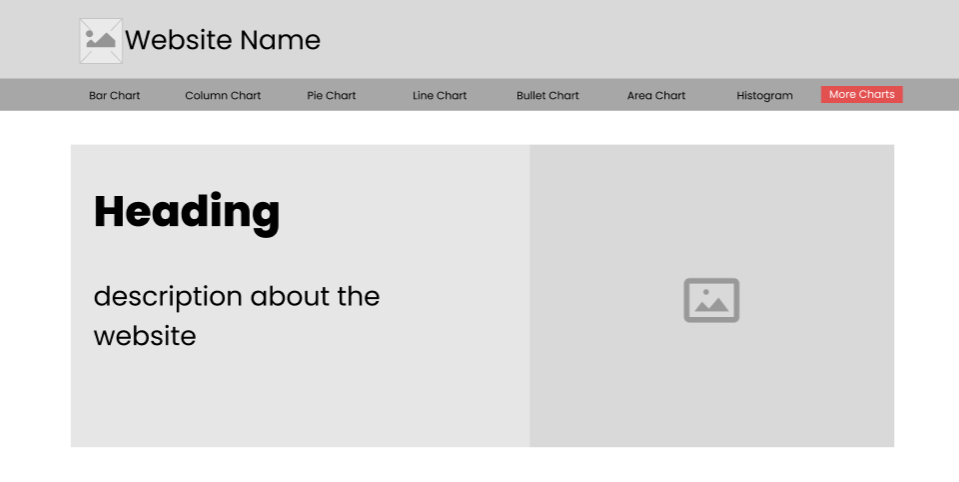
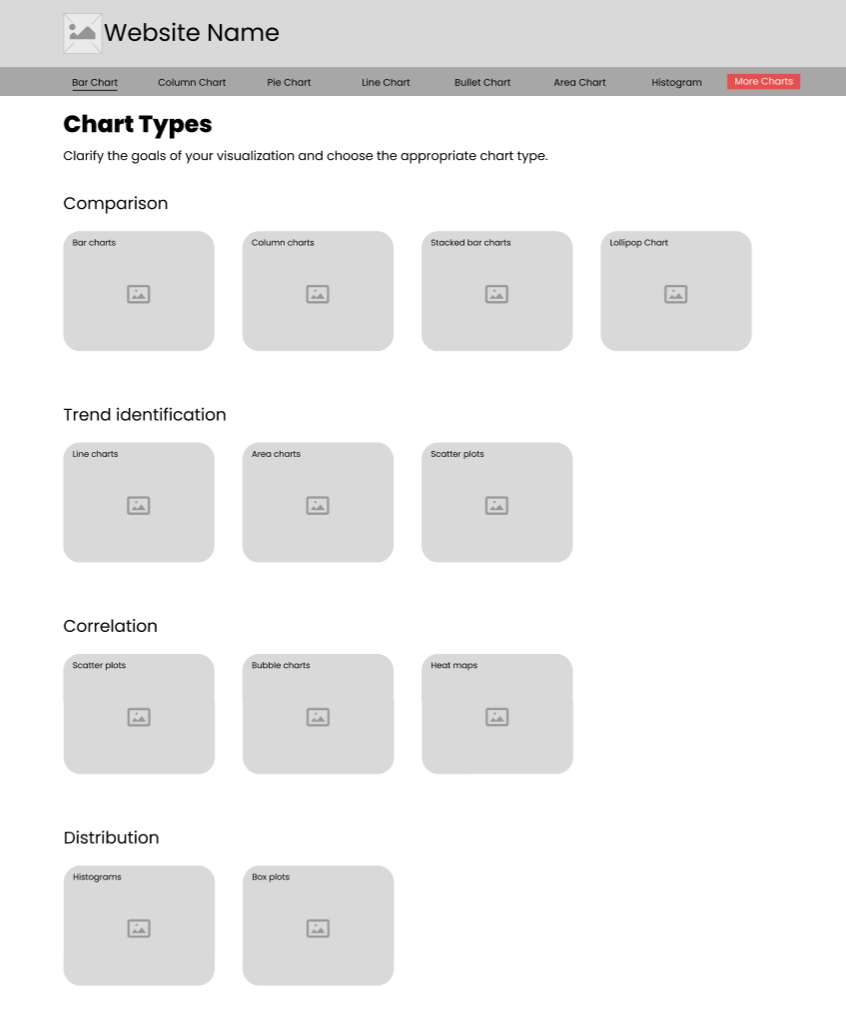


Figure 5.1.2.1 The main page of the application

### 

Figure 5.1.2.2 The main page of a chart



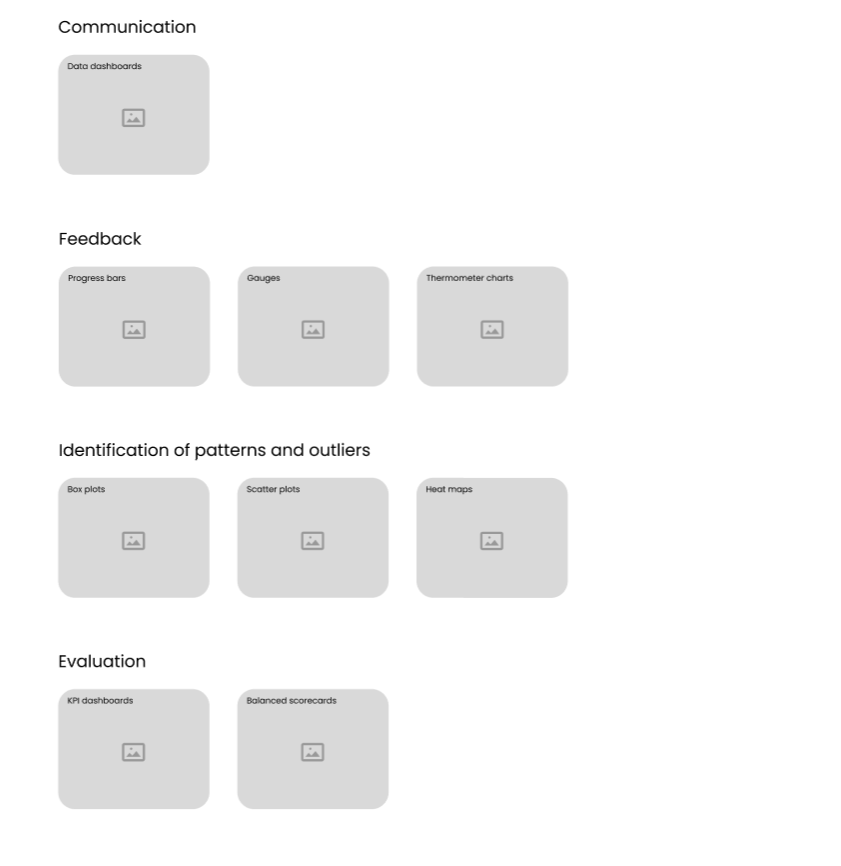


Figure 5.1.2.3 The chart type page

### **5.1.3 Web View(Tablet 768px wide)**

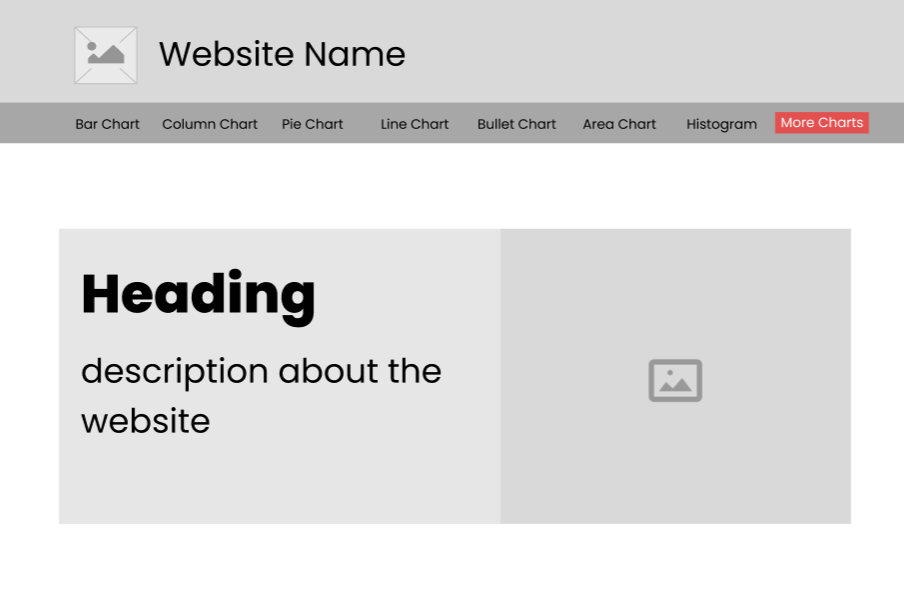


Figure 5.1.3.1 The main page of the application

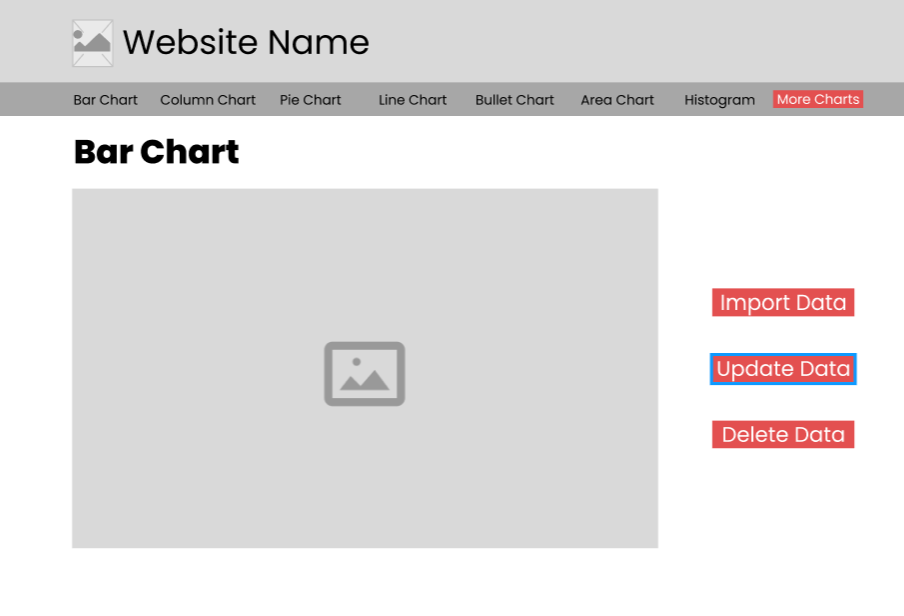


Figure 5.1.3.2 The main page of a chart



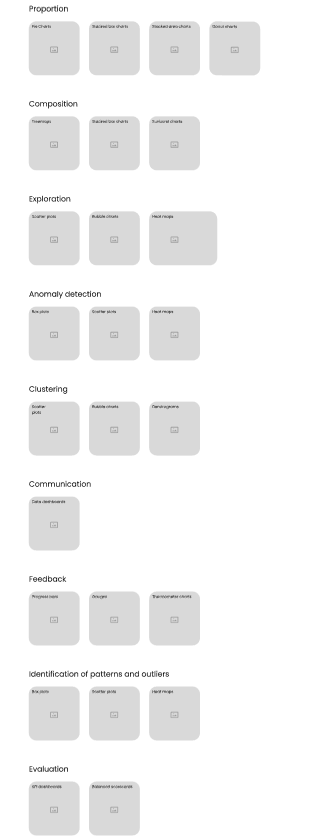


Figure 5.1.3.3 The chart type page

### **5.1.4 Web View(Mobile 320px wide)**

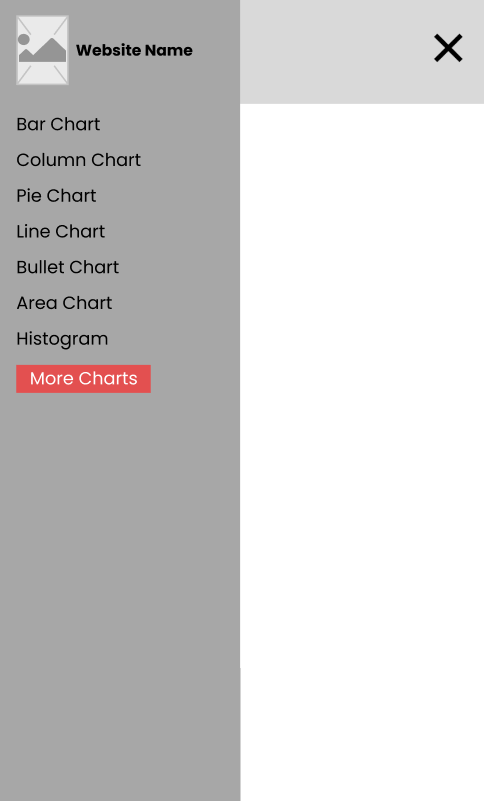
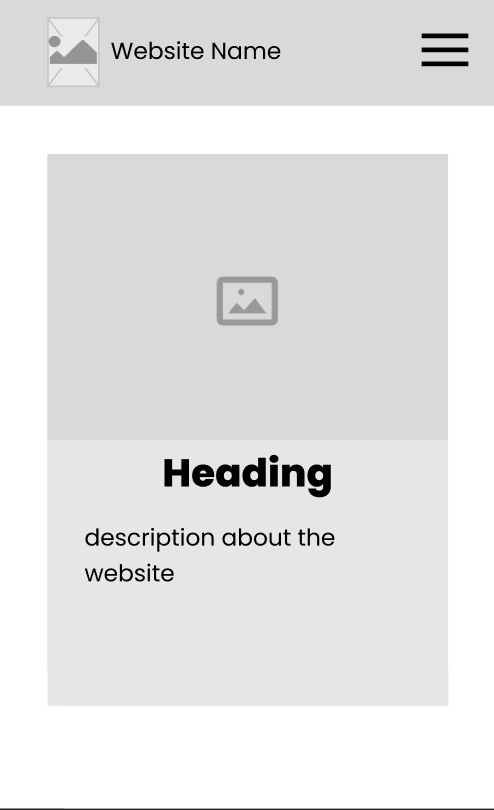


Figure 5.1.4.1 The main page with and without expanded

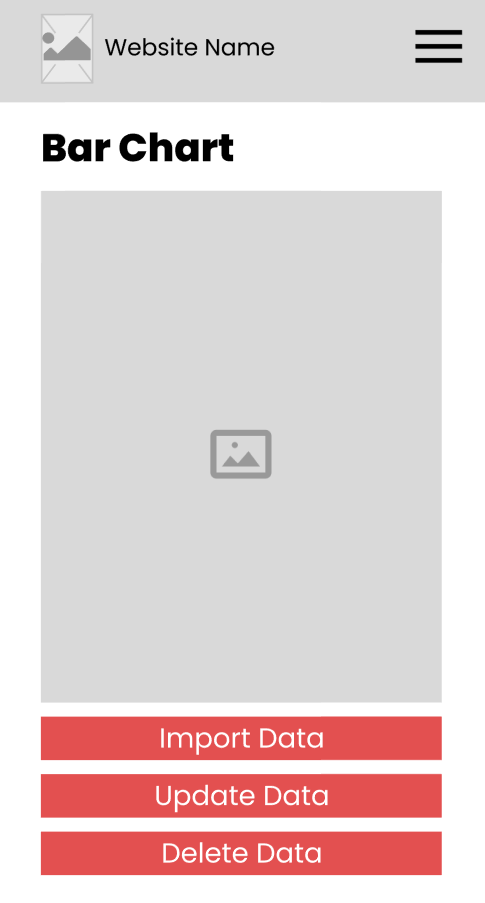
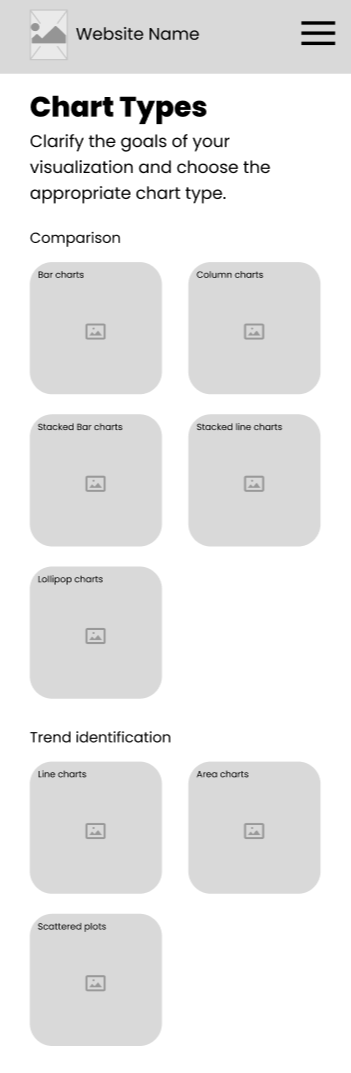
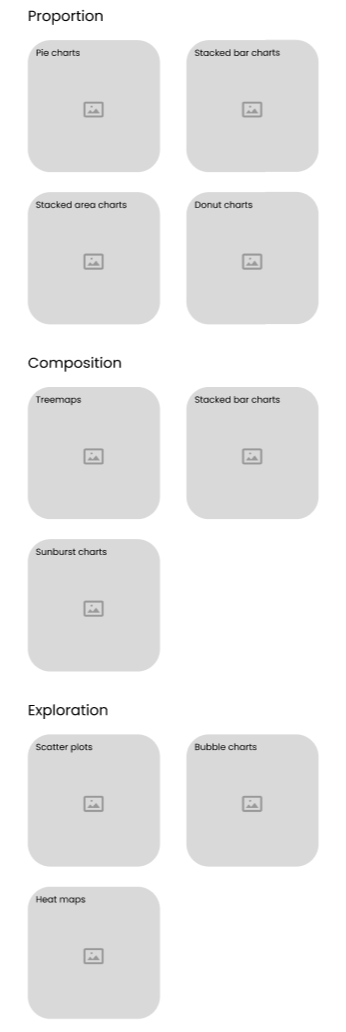


Figure 5.1.4.2 The main page of a chart







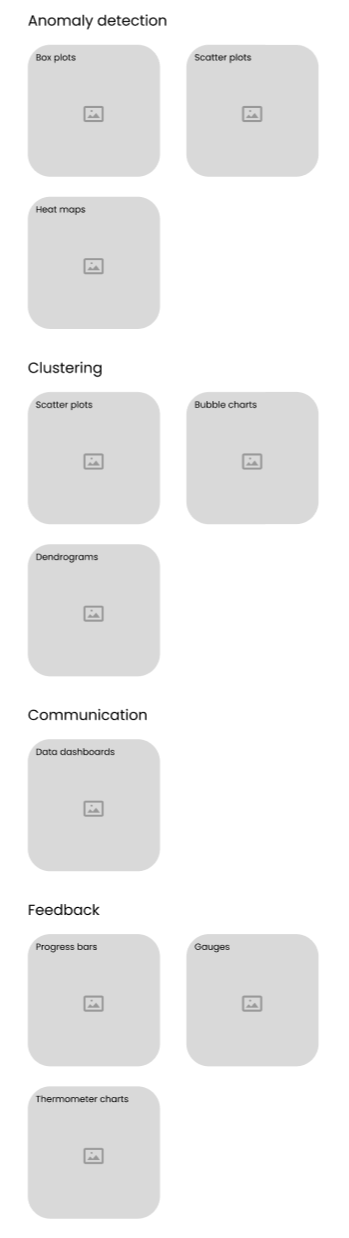




Figure 5.1.4.3 The chart type page

## **5.2 Mockup screen(Wireframe Version 2)**

### **5.2.1 Web View(Monitor 1920px wide)**



Figure 5.2.1.1 The main page of a chart

### **5.2.2 Web View(Laptop 1024px wide)**

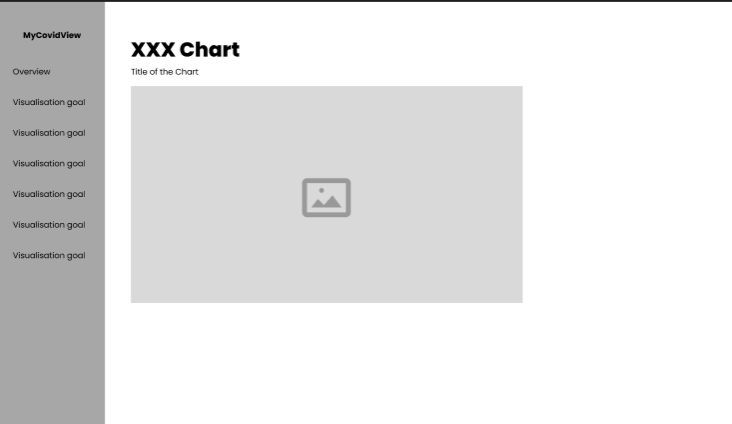


Figure 5.2.2.1 The main page of a chart

### **5.2.3 Web View(Tablet 768px wide)**

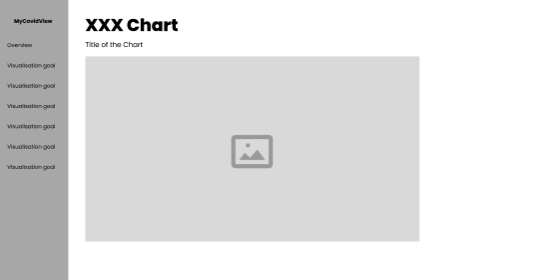


Figure 5.2.3.1 The main page of a chart

### **5.2.4 Web View(Mobile 320px wide)**

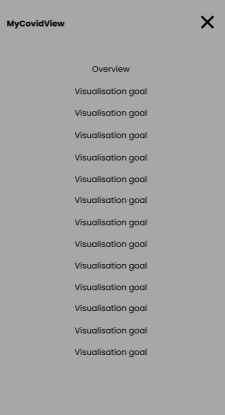


Figure 5.2.4.1 The main page of a chart with and without expanded

## **5.3 Application navigation (flow of the system)**