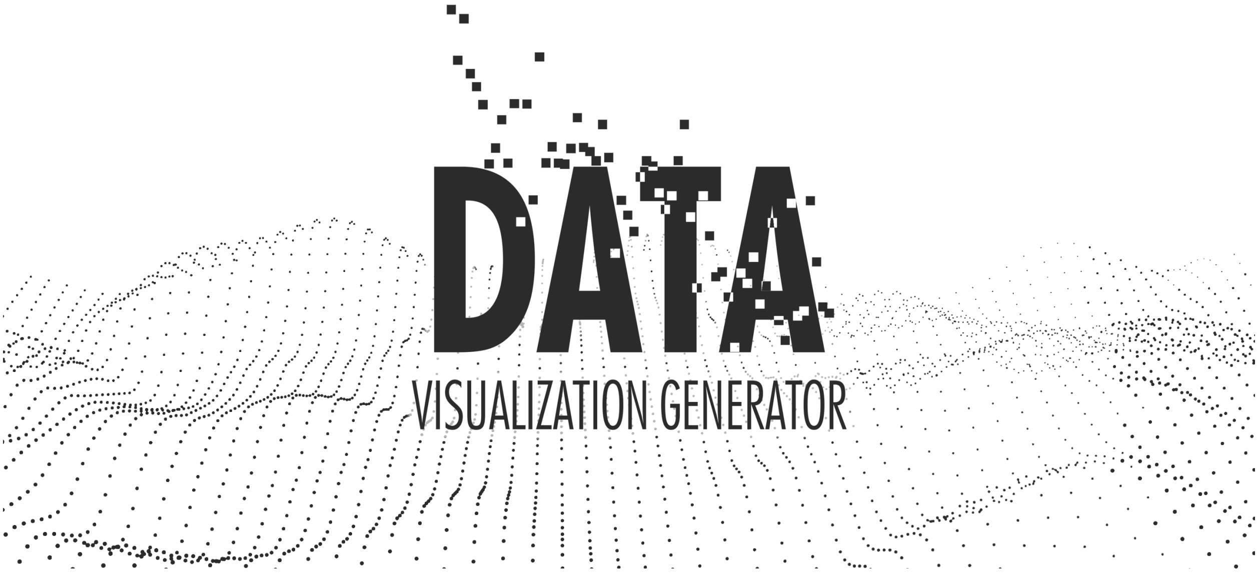
Data Visualization Generator

SRS Document

Doofenshmirtz Evil Inc

COS 301 - 2020

Marco Lombaard u18026975
Elna Pistorius u18010319
Phillip Schulze u18171185
Byron Tomkinson u18042717
Gian Uys u18052569



Contents

1 Introduction

Huge amounts of structured and unstructured data is being stored and processed at a very high rate. This is where the term **'Big data'** comes from. Data is captured to help detect problems and to make better decisions. It is much easier for us as humans to gain insight from data patterns if it is visually represented using charts. These representations take a lot of time to create manually for each data set, especially when we want to create powerful tools like dashboards and drill-downs for end-users.

1.1 Definitions

- **Big data:** It is a field that focuses on extracting information and ways to analyze data sets that are very large or complex to deal with. [1]
- **Charts:** It is a large group of methods for presenting information in the form of graphs, diagrams, or tables.
- **IGA:** (Interactive Genetic Algorithm), is an effective method in solving optimization problems with implicit or fuzzy indices. These types of algorithms combine evolutionary mechanisms with the user's intelligent evaluation and individual fitness. [2]
- **Drill Downs:** This provides the user with the capability to view a more specialized representation of the data. [3]
- **Dashboards:** This is an information management tool used to visually track, analyze, and display key performance indicators (KPI). Dashboards can be customized to meet the specific requirements of the end-user. [4]

1.2 Vision

Data visualization is a progressive web application used to simplify and enhance the way **'Big data'** is being preprocessed and displayed on charts. It takes time to make visualizations manually, instead of building these visualizations from scratch the Data Visualization system will make it easier and faster for the end-user, by providing an intelligent list of auto-generated visualizations as suggestions. These suggestions would be based on data as is or after it is preprocessed through other Artificial Intelligence (AI) algorithms. An **Interactive Genetic Algorithm (IGA)** will be used to suggest visualizations for dashboards and to provide drill-down of these suggestions. These visualizations will help the end-user to make adequate decisions and to make more accurate predictions.

1.3 Objectives

The key objectives that need to be fulfilled so that the Data Visualization system can be used effectively as a tool to visualize data are the following:

- Enable the user to **select a specific data source and trigger visualization generation**. This would be used to provide the user of a visual suggestion of the data. An IGA is used for suggestion generation, the user can then trigger a next-generation calculation if a better suggestion is desired.
- Enable the user to **view and add these suggestions easily to dashboards**. The user can then filter all the visualizations on the dashboard, making it easier to track, analyze and display key performance indicators.
- The Data Visualization system aims to mitigate the time and effort needed for manually creating visual representations of data provided.

1.4 Business Needs

Data Visualization enables companies, governments, and health care providers to generate customized reports and interact with data in an unmediated way.

Data visualization helps these users increase their **productivity** by presenting a visual summary of data, making it easier to **identify patterns and trends**. This could help users spend less time analyzing data but providing a platform where these users can make **informed decisions** based on the visual representations provided. Users can then customize these representations of data to eliminate what they do not require and drill down into more important details.

1.5 Scope

Data Visualization will be a progressive web application, paired with a backend system, this will be accessed through the web interface. Controlled altering and viewing of the data stored on the backend database will be allowed through the backend system and web application. The database will be set up using a database management system like **PostgreSQL**, which is an appropriate relational database management system **(RDBMS)**.To ensure access to this database is secure, a **web API** will be used, this will be implemented using a server-side language such as **NodeJS**. The web application will be developed using a framework such as **React**. End-users will be able to select specific data sources and trigger visualization suggestion generations via the web application. This would send a request to an endpoint on the server that would trigger the IGA to suggest representations. The user must then be able to view and add these suggestions easily to a dashboard. The data sources would be stored in the database on the server and each data entity required from the database would be retrieved via an API. Only administrators would have access to the backend system to be able to alter the database and do any server-side actions.

1.6 Domain Model

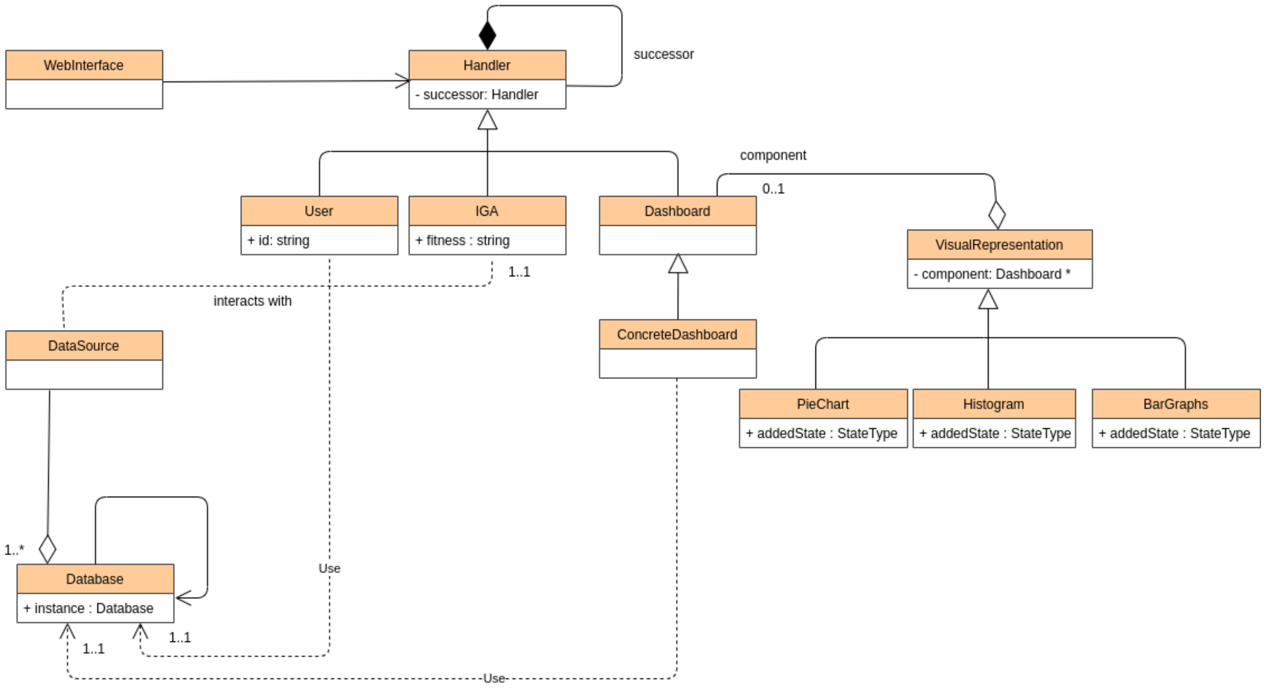


Figure 1: Domain model showing the system components and their relationship with one another

As one can see for the above domain model there are *three design patterns used*, these patterns help to document the architecture of the Data Visualization system and enhances the understanding of the system as a whole. These patterns help to make our code more reusable for future use. The patterns we chose are:

1. **Chain of responsibility:** The **WebInterface** sends requests to the **Handler** and these requests are then 'handled' in a chain of responsibility manner. The requests are sent down the chain to various handling components **(User, IGA and Dashboard)**, and when the requests matches the corresponding handling class, that class will handle the requests appropriately.

   - The **IGA** components will handle all requests with regards to chart suggestions for example when the user selects a different fitness value or selects a data source. The IGA interacts with the **DataSource** to do the processing necessary for the suggestions. The **DataSource** will be stored on the **Database**.
   - The **User** component will handle all requests to be done with a user, for example, creating an account, logging in, saving preferences. This will interact with the **Database** to store or update the necessary attributes.

- The **Dashboard** components will handle all requests done with the dashboard for example creating or saving dashboards. The **Dashboard** must then be stored in the **Database**.

2. **Singleton:** To make the **Database** more secure there is only one instance that will ensure no requests will override each other.

3. **Decorator:** The **Dashboard and Concrete Dashboard** will be decorated by the **VisualRepresentation (PieChart, Histogram, and BarGraphs)**, this allows behavior to be added to an individual object, dynamically, without affecting the behavior of other objects from the same class.

1.7 User Characteristics

These users will use the data visualization system by interacting with the web application. These users will mainly use the system to use the suggested visualizations generated by the IGA, they will use these representations to make informed decisions and identify patterns in the data presented.

- **Data Scientist:** The most frequent user of the system would be data scientists and analysts. As they are tasked with interacting with the data in order to derive further conclusions.
- **Business Analyst:** These analysts are responsible for providing their respective companies with data-driven recommendations. Their entire investigation into the requirements of a software system makes use of data and would require a visual tool for representing.
- **Casual User:** Novice users with some interest in Big Data and Data Science. These types of users would like to explore this subject purely for pleasure. They would not particularly create personalized dashboards, even though they have the functionality to do so if they desire to, they will mostly use the system just to look at charts and make assumptions.

1.8 User Stories

A user story is accompanied by lots of other documentation – these user stories are just a visual representation used for planning and to help setting up the product backlog.

1. **As a** business analyst, I want to provide the system with data concerning my respective company's financial data **so that** I can determine possible wastefulness of assets.
2. **As a** business analyst, I want to organize my dashboard with charts **so that** I can easy tell a story about the data.
3. **As a** business analyst, I want to store my dashboard creation **so that** I can present the dashboard to my colleagues, superiors, and clients.
4. **As a** business analyst, I want to select only a certain amount of fields from a data source **so that** I can interpret the charts in a more concise and informative manner.
5. **As a** casual user, I want to be able to view template dashboards **so that** I can make assumption about the data.
6. **As a** casual user, I want to be able to edit template dashboards **so that** I can personalize dashboards to my liking.
7. **As a** data scientist, I would like to drill-down into specific suggestions **so that** I can make accurate forecasts about the stock market.
8. **As a** data scientist, I would like to make a dashboard about risk factors for driving **so that** I can use suggestions to accurately apply insurance policies to specific people.
9. **As a** business analyst, I would like to select a template dashboard **so that** I can quickly get a grasp of the functioning of the system.

1.9 Product Backlog

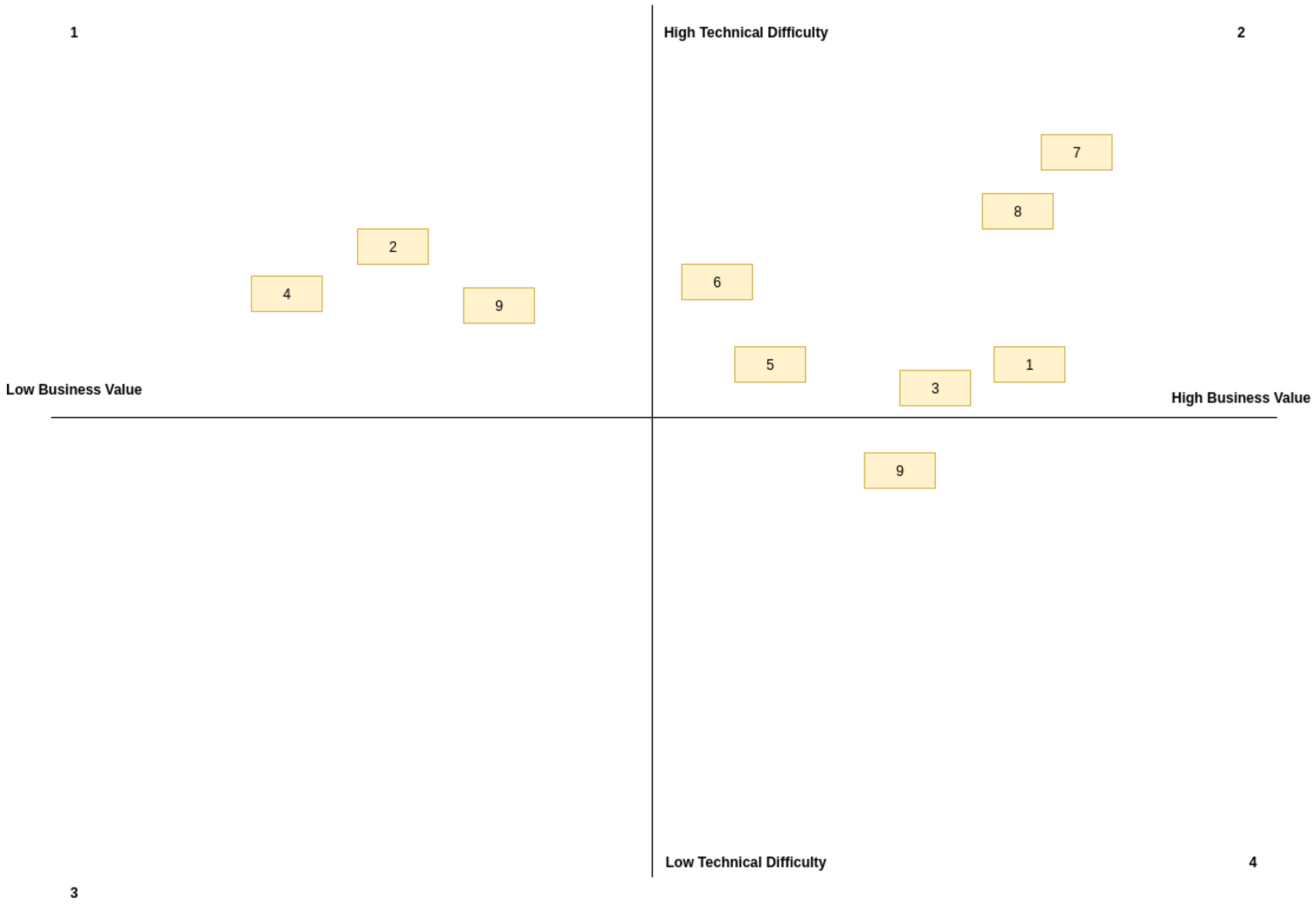


Figure 2: Product Backlog diagram showing the user stories

2 Functional Requirements

2.1 Use cases

Actors - User (Business Analyst, Data Scientist, Casual User)

User Interaction Subsystem
UC01 Create a user profile (Actor: User, System: Web Interface)
UC02 Log in to Data Visualization system (Actor: User, System: Web Interface)
UC03 Select a data source (Actor: User, System: Web Interface)
UC04 Enter data set URL (Actor: User, System: Web Interface)
UC05 Select existing data source (Actor: User, System: Web Interface)
UC06 Select chart types (Actor: User, System: Web Interface)
UC07 Create dashboards (Actor: User, System: Web Interface)
UC08 Configure dashboard layout (Actor: User, System: Web Interface)
UC09 Personalize dashboard (Actor: User, System: Web Interface)
UC10 Open dashboards (Actor: User, System: Web Interface)
UC11 Edit dashboard (Actor: User, System: Web Interface)
UC12 Name dashboard (Actor: User, System: Web Interface)
UC13 Describe dashboard (Actor: User, System: Web Interface)
UC14 Save dashboards (Actor: User, System: Web Interface)
UC15 Add charts to a dashboard (Actor: User, System: Web Interface)

UC16 Select fields from data set (Actor: User, System: Web Interface)
UC17 Add filters to charts (Actor: User, System: Web Interface)
UC18 View template dashboards (Actor: User, System: Web Interface)

Interactive Genetic Algorithm
UC19 Trigger IGA to give chart suggestions based off of data set chosen (Actor: User, System: IGA)
UC20 Update IGA fitness. (Actor: User, System: IGA)

2.2 Use Case Diagram

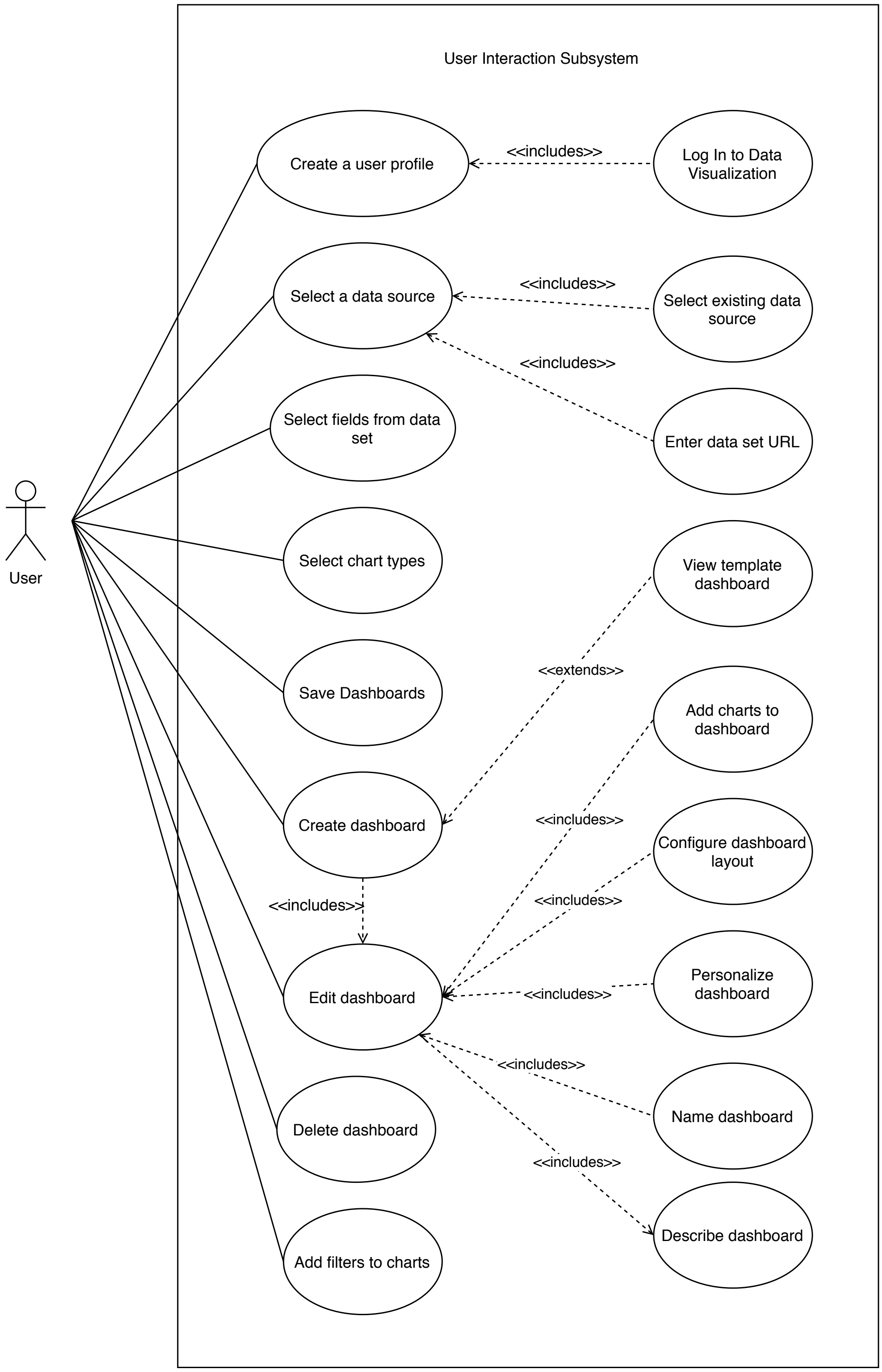


Figure 3: Use Case diagram showing the use cases surrounding the *User Interaction* Subsystem

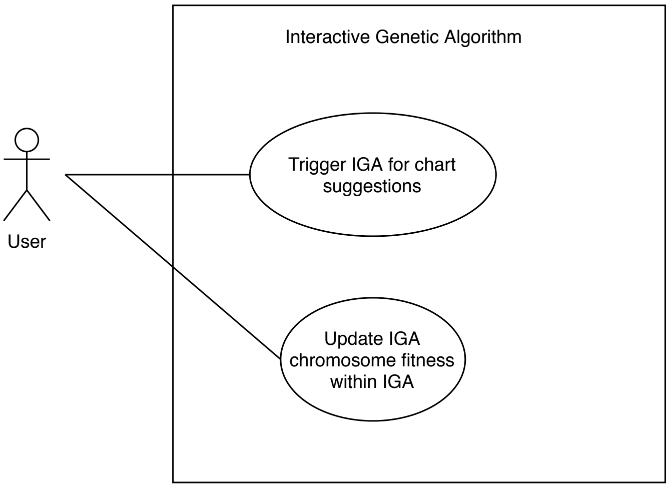

Interactive Genetic Algorithm
User
Trigger IGA for chart suggestions
Update IGA chromosome fitness within IGA

Figure 4: Use Case diagram showing the use cases surrounding the *Interactive Genetic Algorithm* Subsystem

2.3 Functional Requirements

R1 The Data Visualization system must be able to let users create an account.

R1.1 The system must be able to retrieve all of the user's dashboards that they created or saved.

R1.2 The system must be able to let users retrieve all their preferences, such as chart types and filters selected (if any is present).

R1.3 The system must be able to retrieve template dashboards that is provided by the system.

R2 The Data Visualization system must be able to let users select specific data sources that trigger a visualization suggestion generator.

R2.1 The system must be able to provide the user with suggestions of that specific data source in the form of a visual representation via a chart etc.

R2.2 An **IGA** is used for the suggestion generation. When drilling into these suggestions it uses the fittest individual and triggers the next generation of suggestions. These suggestions are used to indicate some kind of resemblance/property from the fittest suggestion of the previous generation.

R2.3 The system must be able to let a user view and add these suggestions easily to a dashboard.

R2.4 The system will also allow users to specify what type of fields they want in the chart and this will then provide a chart that reflects these fields selected.

R3 The Data Visualization system must be able to let users create/save multiple dashboards.

R3.1 The system will allow users to give the dashboard a specific name and also a description of what the dashboard is about.

R3.2 When the user views the suggestions provided by the system, the user can add these suggestions to dashboards. These dashboards are then customized to the specific user to visually track, analyze, and display key performance indicators.

R3.3 Users can look at suggested dashboards provided by the system and may save these dashboards for use in the future.

R4 The Data Visualization system must be able to let the user selects a specific chart type.

R4.1 When the user selects specific chart types then only that type needs to be used to create suggestions, otherwise, all types must be considered.

R5 The Data Visualization system must be able to let the user select different fields.

R5.1 When the user selects different fields, then only these fields need to be used by suggestions, otherwise, all the fields and their data types must be considered.

R6 The user needs to be able to add filters and when used they should filter all the visualizations simultaneously on the dashboard.

R6.1 When the user adds a filter on the data set for example choosing a certain time frame, then all the visualizations should update simultaneously.

R7 The progressive web applications interface should be responsive.

R7.1 The interface should adjust to different screen sizes, including mobile phones.

2.4 Subsystems

S1. Data Storage and Organisation Subsystem

Description

This subsystem is responsible for maintaining the database and performing **CRUD** operations on the database. It will the interface for database operations to all other subsystems.
Related Functional Requirements:

- R1.1
- R1.2
- R1.3
- R3.1
- R3.3

S2. IGA subsystem

Description

This subsystem is responsible for generating suggestions for graphs when it is triggered. This is triggered via a user using the web interface and must generate chart suggestions based off the given data source and fields.
Related Functional Requirements:

- R2.1
- R2.2
- R3.2
- R4.1
- R5.1

S3. Dashboard Subsystem

Description

This subsystem is responsible for handling any operations that is done to a dashboard, including creating, editing, saving and other operations. Dashboards are the main way for a user to see the visual representations of the data sources and are viewed on the web interface.
Related Functional Requirements:

- R2.3
- R2.4
- R3.1
- R3.2
- R3.3
- R6.1

S4. Progressive Web Application Subsystem

Description
This subsystem is responsible for the web interface, this will therefore handle any interaction between the user and the system as the web interface is the main means for a user to use the Data Visualisation system.
Related Functional Requirements:

- R2.1
- R2.3
- R2.4
- R3.1
- R3.2
- R3.3
- R4.1
- R5.1
- R6.1
- R7.1

3 Quality Requirements

The requirements in this section provide information about the quality of the application and what the application should be able to achieve. These are the **functional quality requirements**, the processing capabilities that the software system **must** posses. The non-functional requirements will be discussed later, as we progress with our project.

Q1. Performance

- The Data Visualization system must be able to handle 180 requests per second.
- The system must be able to respond to an initial request in under 10 second but will depend on internet connection strength and location of the server.

Q2. Reliability and Availability
The Data Visualization system will be hosted on an external server with a contractual agreement with the service provider.

1. **Reliability:**
   - The Data Visualization system must be tested before deployment and the system must behave the same in deployment as it did in testing.
2. **Availability:**
   - The Data Visualization system must available for 98.9% throughout its lifetime.
   - The Data Visualization system must have access to the databases for 98.9% throughout its lifetime.

Q3. Scalability

- The Data Visualization system must be designed using appropriate design patterns to allow for easy scalability of the functionality of the system.

Q4. Security and Maintainability

1. **Security:**
   - **Data Storage:**

  - The data of users must be stored in a secure manner and must have controlled access.
  - All data that conform to the Customer Laws need to be logged for the required amount of time and must be deleted after a certain amount of time.
  - Audit logs must be stored and must only be accessible to the product owners.
  - Client passwords must be hashed and salted before storage.
- **Data Transfer:**
  - Data sent over the internet must be encrypted and securely transferred between different locations.
  - When sending data, a user must be prompted to transfer the data or deny the transfer of data.
- **Data Access:**
  - All data must have clearance levels associated with it, which will give controlled access to data.
  - All data logs must have controlled access and can only be accessed through an interface (not API), only users with desired clearance levels may access the data.
  - The owner of the product must be able to add privileges or remove privileges from a client.

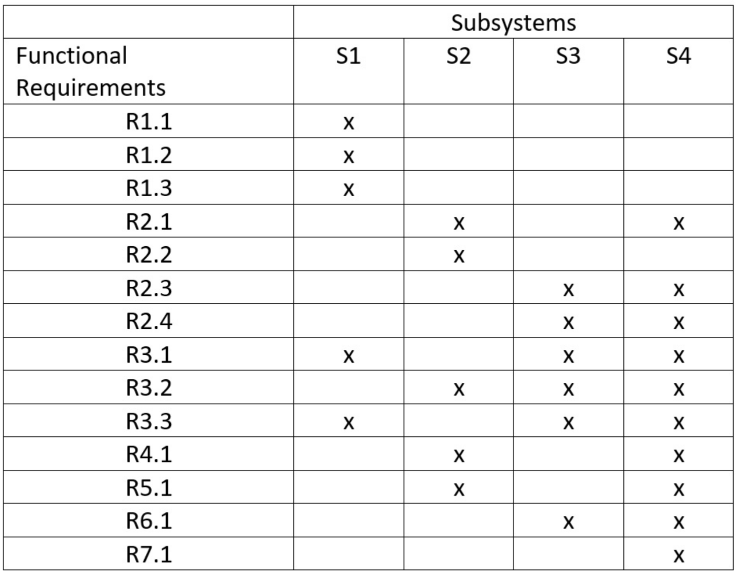
2. **Maintainability:**
   - To ensure our system is easily maintained, a modular approach would be followed. This approach is know as **Modularization**, this principle facilitates the functionality of low coupling and high cohesion. This means the system must be sub-divided in to smaller systems that make maintenance and updates to the system easier

Q5. Usability

- The Data Visualization system must have an easy to navigate user interface to allow all users to understand the application.
- The Data Visualization system must be designed in a vertical approach rather than a horizontal design to make navigating through the application easier.

The user-interface must provide the user with a good user experience **(UX)**, a user-friendly product is in general more successful and users will find the system more reliable if the system is self-explanatory. This is an ideal situation as this means our **goal of developing a successful product is met**.

4 Traceability matrix



	Subsystems			
Functional Requirements	S1	S2	S3	S4
R1.1	x			
R1.2	x			
R1.3	x			
R2.1		x		x
R2.2		x		
R2.3			x	x
R2.4			x	x
R3.1	x		x	x
R3.2		x	x	x
R3.3	x		x	x
R4.1		x		x
R5.1		x		x
R6.1			x	x
R7.1				x

Figure 5: Traceability matrix

References

[1] Min Chen, Shiwen Mao, and Yunhao Liu. Big data: A survey. *Mobile networks and applications*, 19(2):171–209, 2014.

[2] Dun-wei Gong, Xiao-yan Sun, and Jie Yuan. Interactive genetic algorithms with individual's uncertain fitness. *Evolutionary Computation*, pages 21–44, 2009.

[3] Jacob Andrew Taylor, Majed Itani, Ajay Gupta, Andrew Wu, Joseph Parsons, Roger Smith, and Chris Nojima. Crm system and method having drilldowns, acls, shared folders, a tracker and a module builder, March 12 2009. US Patent App. 12/200,301.

[4] Lidong Wang, Guanghui Wang, and Cheryl Ann Alexander. Big data and visualization: methods, challenges and technology progress. *Digital Technologies*, 1(1):33–38, 2015.