CA400 - Final Year Project

Functional Specification for

DATATIVE

by:

Camilla Boyle - 16732949 and Emily McGivern - 16305506

Date finished working on the document:

22nd November 2019

Functional Specification Contents

1. Introduction

1.1 Overview

Datative is a web-based analytics and dashboarding tool. It gives users full control of their data and allows them to visualise their data in a way that is meaningful to them. It is a rich, personalised analytics tool which offers full customisation to the user. It will provide a user numerous ways to intuitively visualise and understand the outcome of the analytics that they have performed on their data. It will also allow them to export their dashboard in PDF format and save reports for future use. The need for this system stems from lack of available analytics and dashboarding tools available that cater for a user which little to no knowledge of performing analytics and designing analytics dashboards. It does not require the connection of a data source but instead allows the user to upload their data in various formats. It also does not require knowledge of analytics as all the algorithms provided will be clear and provide useful information to the user.

The user will upload their data to the tool using an upload button. The data can be uploaded in a number of different formats, such as CSV and text files. The data will then be cleaned and stored in the database. The user can choose what data that they want to add to various components on their dashboard. Once they have chosen their component, they can move it anywhere on the dashboard canvas. The components, along with the dashboard, will be fully customisable. Users can choose the colours of their components, along with adding custom labels to components. They will also be able to add headers and footers to the canvas. Once they are happy with the results, they can export their dashboard in the form of a PDF report. They can also save their report in the database and retrieve it in the future.

1.2 Glossary

1. **Spring Boot -** open source Java-based framework for rapid application development
2. **Google Cloud Platform -** Suite of cloud computing services offered by Google
3. **Cloud Storage Bucket -** Basic container to hold data
4. **jsPDF -** HTML5 client-side solution for generating PDFs
5. **Cloud Dataprep -** intelligent cloud data service to visually explore, clean, and prepare data for analysis

2. General Description

2.1 Product / System Functions

The general functionality of this web application is to allow users to generate meaningful analytics and reports from their data. The user will be able to perform a wide range of analytics on their data which will cover their business and personal needs. The user will be able to choose which areas of their data they wish to target and analyse. They are also not restricted with what forms of data they can upload to

the application as it will be cleaned and stored in JSON format in the database. The application will also store the data so that the user does not have to repeatedly upload the same dataset to the application. The data will be stored for 3 months so in that time the user can work with their data as much as they want. If they want to work with it again in the future, they can upload the data again. The application will offer a wide range of algorithms which can be performed on their data. This will allow users to gain meaningful insights into their data. Examples of algorithms offered in the application are:

1. Sum
2. Time Series
3. Aggregation
4. Clustering
5. Trend Analysis (using leading and/or lagging indicators)

The resulting analytics will be displayed in components which the user can choose. Example components which will be included in the application are bar charts, line graphs, KPI tiles and crosstabs. This gives the user a wide range of visualisations to include in their dashboard and supports the user in creating any sort of dashboard that suit their needs. The dashboard can be further customised with a number of other features, such as custom background colour, inclusion of headers and footers and metadata. Once the user is happy with their dashboard layout, they can export it as a report in PDF format. They can also store their report to return to it in the future, and can give it a custom name.

2.2 User Characteristics and Objectives

This tool will be a web-based application. Therefore, it can be accessed by anyone with an internet connection. The target audience is broad and encompasses many different disciplines. The tool can be used by almost any employee in a business. This means, from a sales manager needing to view sales over a certain period of time, to a marketing manager who wishes to analyse the demographics of the customers they deal with. Other applications are it can be used in education to analyse student performance in exams, or in sports to analyse an athletes performance. The main type of user we hope to target is one who has a need for further analysis of their data, without purchasing outside expertise or expensive packages, such as those currently available.

It will be an easy-to-use tool and will target users of all skill levels instead of just professional analysis applications designers. To achieve this, we must take the interface into account. Users will not need extensive knowledge to use this tool. The interface will be simple and tidy. The functionality of the buttons will be clear and obvious. The data uploaded by the user will be displayed in simple table formats so that they can choose the corresponding rows and columns they want to use in a component. The components will not be over-complicated but will display the data to the user in an interesting fashion. There will be clear boundaries which show the location of the canvas and where the user can drag their components. The objective of this application is to allow any user to gain insight from any type of data. It will allow users to see their data in very different ways.

2.3 Operational Scenarios

Use-Cases

2.3.1 Register with Email Address

Goal in context: User successfully registers themselves on the website

User accesses website by entering the address into their chosen browser's address bar. When the website has loaded, the user will be presented with a screen where the user can log in or register. If the user has not previously used the website, the user will click the register button.

There will be two options on this page. Either the user can register with an email address and create a username and password, or they may register with a Google account. If registering using a Google account, the user will click on the button labelled "Continue with Google".

If the user decides to register with an email address, they will be prompted to enter this email address into a text box along with username and corresponding password in another text box. The user will then be asked to confirm their password by entering it again into a text box.

A button labelled "register" will be underneath these options. If the passwords match, a message accepting registration will be displayed. The user will then be redirected to the login page. If the passwords do not match, the user will be instructed to re-enter their password in the corresponding two text boxes. If the email address is already attached to a user, an alert will be displayed and the user will be redirected to the login page where they can choose to log in or reset their password.

2.3.2 Register with Google

Goal in context: User successfully registers themselves using Google account

Should the user decide to register with a Google account, they will click on the button labelled "Continue with Google". They will then be redirected to the Google login page where they will enter their Google account details and click the "sign in" button. The user will then be logged in and redirected to the user's home page on the application.

2.3.3 Log In

Goal in context: User successfully logs into website

User accesses website by entering the address into their chosen browser's address bar. When the website has loaded, the user will be presented with a screen where they can log in or register. If the user has previously made an account, the user will log in using their email address and password or their Google account. There will be two buttons will be on the screen below these text boxes, they will be labelled 'Forgot Password' and 'Log In'.

If the user selects the 'Forgot Password' button, they will be asked to enter the email address used to create the account in a text box. They will click the confirm button. Upon clicking this button, they will receive an email with a link to reset their password. Once they reset their password, they will be redirected to the Login screen.

If the user tries to log in and their username and password match, the users home screen will appear. If the user tries to log in and their username and password do not match, the screen will display an alert asking them to re-enter their details in the text boxes.

2.3.4 Upload and Save File

Goal in context: User successfully uploads a file to the application which is saved to the user profile

User clicks the "Upload File" button. A window will open offering the user the ability to select from their file directory. Alternatively, the user can drag a file from their desktop and drop it into the upload window. The user will then select the "upload" button and the selected file will be uploaded. A window appears confirming the data has been uploaded.

This data is saved after cleaning and processing. It will be displayed in the user's profile page.

2.3.5 Data Selection

Goal in context: The data the user wishes to visualise has been selected

The user selects the data they wish to analyse by double clicking on the data that has been uploaded. The data will then appear in the dashboard window. From here, the user will select the specific rows, columns and/or attributes that they wish to focus on by clicking the tick boxes on the left hand side of the window. From here, they will select the components and parameters required.

2.3.6 Component Selection

Goal in context: The component the user wishes to use on their data has been selected

When the data has been selected, the user can then decide how exactly they would like their data to be displayed. From a menu bar, the user can select several different options, such as line graphs, scatter graphs, bar charts, KPI tiles and crosstabs. Unsuitable components will be greyed out, for example, a pie chart will not be proposed to display time series data. The selection will be displayed on the canvas. The user can drag and move these displays around the canvas at will using the mouse in the traditional “drag and drop” manner.

2.3.7 Parameter Selection

Goal in context: The user has set the parameter or parameters for their analysis

When the data has been selected, the user can then decide whether to set parameters on the data. For example, should it be sales data with a date, they could select whether to view the data by week, month, quarter, or year. Similarly, it would be possible to view sales data by product, location or price, should these be labelled in the data.

The various filters will be data-dependant and will be in the form of tick boxes along the side of the dashboard. The user will select the option or options using the tick boxes and click “apply filter” button. The component will then display the data according to the filters applied.

2.3.8 Rearranging Components

Goal in context: The user will rearrange their dashboard

When the user is satisfied with their component, they can place it in the desired area on the canvas. This is done by dragging and dropping the component to an area within the dashboard. The boundaries of the canvas will be clearly visible to the user. Once the user drops the component within a certain area, the component will remain there until it is moved again.

2.3.9 Export PDF

Goal in context: The results of the dashboard are exported as a PDF

Once the user is satisfied with their analysis and the placement of the components, the user may wish to export these results. From the dropdown menu titled “File” in the dashboard, the user selects the “export as PDF” option. A window will appear with the option to name the file that the user

wishes to export. The user will enter the title of the file in a text input box. The user will then click the “export” button below the text input box. A confirmation window will appear and the file will be exported.

2.3.10 Save PDF

Goal in context: The results of the dashboard are saved to the user profile

After the user has exported their report as a PDF, they will be given the option to save a copy of the report in the application database. On the confirmation window that appears after exporting, the user clicks the button “Save Report”. The user will enter a title for the report they are saving to the database and click “Save”. The report will be visible in the user profile page where they can click to view it.

2.3.11 Log Out

Goal in context: The user successfully logs out of website

The user may log out at any point by going to the dropdown menu that is visible at the top bar of the website, and selecting “Log Out”. A window will appear with the option to confirm that the user wishes to log out and leave the site. The user will click “log out”. A confirmation window will appear. The user will then be redirected to log in/register page.

2.4 Constraints

2.4.1 Testing Constraints

We must ensure that we have suitable datasets that we can use to test the application

2.4.2 Time Constraints

We must ensure that we stick closely to our planned timeline to ensure that we have the project completed before the deadline as we must balance our project work with our work for other classes and exams.

2.4.3 Billing Constraints

We have a certain amount of free credits which we can use for testing purposes on Google Cloud Platform. We must be careful we do not exceed these free credits as running the application for testing purposes would be costly otherwise.

2.4.4 Useability Constraints

Due to the fact that we are aiming our tool at users with as rudimentary working knowledge of data analysis, we must ensure that our finished product is extremely user friendly.

3. Functional Requirements

3.1 User Registration

Description - The user must create an account. When registering, they must supply an email, username, password and re-entry of the password. These details are then stored in the database. The user can then continue to log in with these details in the future.
Criticality - Due to the requirement to store data and reports associated with an individual user, this requirement is completely essential to the system.
Technical issues - Login information is extremely sensitive. The primary concern with storing this data is security. The log in details must be stored securely, and the password cannot be stored in plain text as this heightens the risk of a security breach. The password must be hashed before being stored in the database. The password will not be accepted unless deemed to be complicated enough to be secure.
Dependencies with other requirements - This requirement is dependent on the database as the user details are stored here

3.2 Log In

Description - The email, username and password supplied by the user during registration is stored in the database to be used for log in
Criticality - The login feature is essential for the user to be authenticated. It also ensures the stored data and reports shown to the user is the data which is related to the user currently logged in.
Technical issues - The log in details must be unique to each user to ensure that there are no duplicate log in details stored. Security must be in place to prevent unauthorised access to the system.
Dependencies with other requirements - This is dependent on the user registration requirement as the user must have registered to log in.

3.3 File Uploads

Description - The user must upload a file containing the data which they want to be analysed and displayed within the application. Once the file has been uploaded once, the data is then stored in the database for future use.
Criticality - Without a file uploaded by the user, there will be no data to analyse and create dashboards with.
Technical issues - The file that is uploaded must contain data suitable to be analysed, and there must be data which can be read off the file.
Dependencies with other requirements - Cloud Storage Buckets

3.4 Data Cleaning and Processing

Description - The data must be cleaned and processed in order for it to be possible to perform an analysis on it. The data will be cleaned by Cloud Dataprep which will return the cleaned and prepared data to the Cloud Storage Buckets.
Criticality - Without clean, processed, useable data, the results of the analysis risk being inaccurate or insufficient. If the data can not be cleaned, analysis will not be possible.
Technical issues - The file that is uploaded must be in a format that is recognised. The data should be easily converted into a format that is suitable for data cleaning and processing.
Dependencies with other requirements - Cloud Storage Buckets, File Uploads/Data Storage

3.5 Data Selection

Description - After the data has been stored in JSON format, the user can select what elements of the data they wish to analyse.
Criticality - It is vital to allow the user to select the type of data that they wish to analyse. This is one of the core functionalities of the tool.
Technical issues - The data must be in a format that allows the system to recognise how and where the data is separated. Should the data be in the format of time, this should be easily identifiable. It should be easy for the end user to identify the different options available to them.
Dependencies with other requirements - Components and Visualisations, Cloud Storage Buckets, Drag and Drop Canvas Area

3.6 Analytics

Description - After the data has been stored in JSON format, the user can choose which algorithm they want performed on the part of data they have chosen. The data is then transformed using the algorithm.
Criticality - The analytics generated using the algorithms are what will be used to create the components
Technical issues - The resulting analytics must be displayed in the correct type of component to ensure they will have meaning to the user
Dependencies with other requirements - Components and Visualisations, Data Selection

3.7 Components and Visualisations

Description - When the data has been selected the user can choose which components and visualisation to use with their data. The visualisation is then displayed on the Drag and Drop Canvas Area. The tool to create the visualisation will depend on the component. Examples of visualisation tools used are chart.js and react-vis.
Criticality - As our tool's main feature is the visualisation and reporting of data, this is a vital component of the app. Without the components and visualisations, we cannot display the results to the user.
Technical issues - The data must be displayed in a way that allows a better understanding of the data. It should be clear and in a format that is easy to use. The unsuitable components should be greyed out where necessary.

Dependencies with other requirements - Analytics, Cloud Storage Buckets, Data Selection, Drag and Drop Canvas Area

3.8 Drag and Drop Canvas Area

Description - The user can position their chosen components anywhere on the canvas area within the dashboard
Criticality - The core functionality of the tool requires that a user can create fully customisable dashboard so therefore they must be able to move components
Technical issues - The components cannot overlap and must be resizable so that they fit in the canvas area
Dependencies with other requirements - Components and Visualisations

3.9 Generating a Report

Description - Creating a PDF report of the dashboard that the user has created using jsPDF
Criticality - The core functionality of the tool requires that a user can generate reports and therefore is required
Technical issues - The generated PDF must match the dashboard exactly as the user has created it so all components and elements must be exported correctly
Dependencies with other requirements - Components and Visualisations

3.10 Database

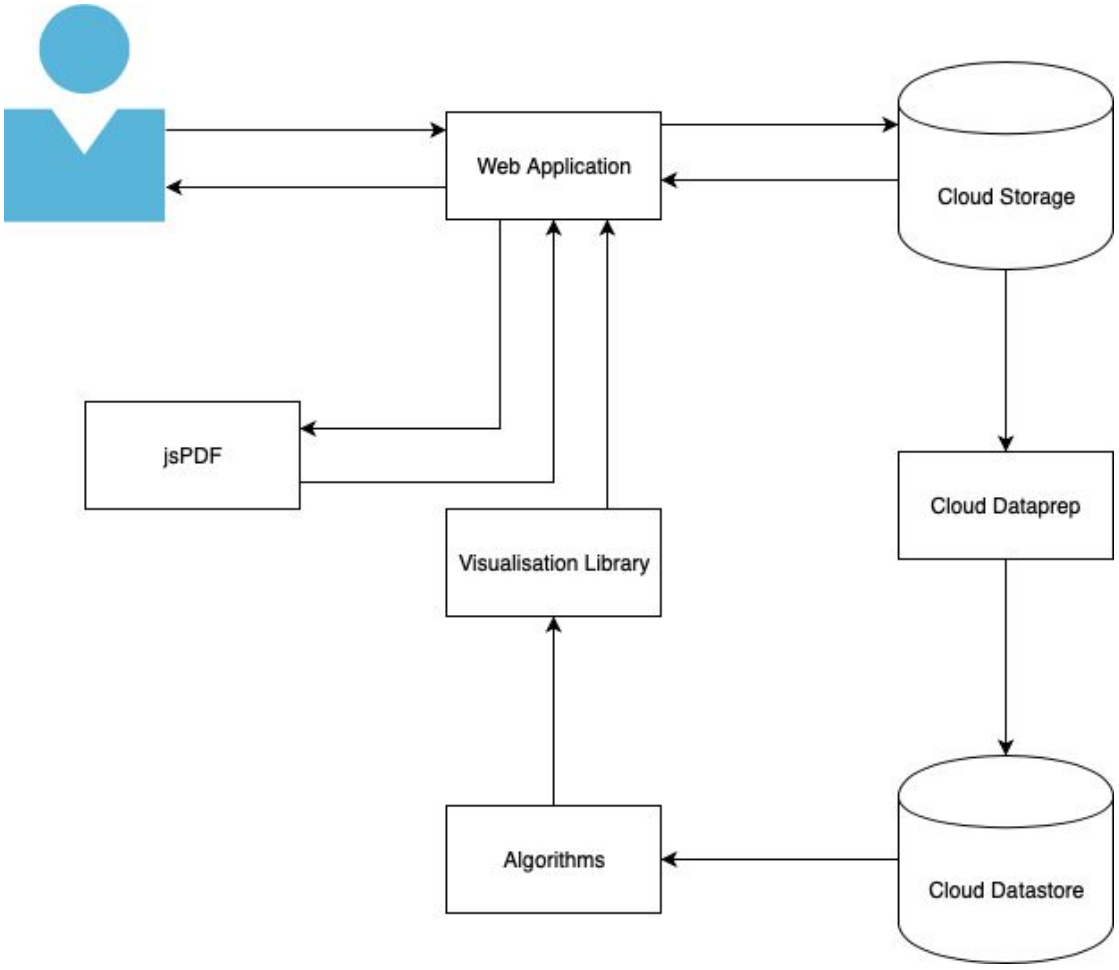
Description - The database will store the user log in details, and the data which is collected from the files uploaded from the user. The user password will be hashed, while their email will be stored in plain text. The data from the uploaded files will be stored in JSON format.
Criticality - In order to perform analytics on the data, it must be stored safely and securely in the database. User details must also be stored so that they can be authentication for each session
Technical issues - The data must be stored safely and we can only store data for a certain period of time before it is removed to prevent the database becoming too full with redundant data
Dependencies with other requirements - File Uploads, User Registration, Log In

3.11 Cloud Storage Bucket

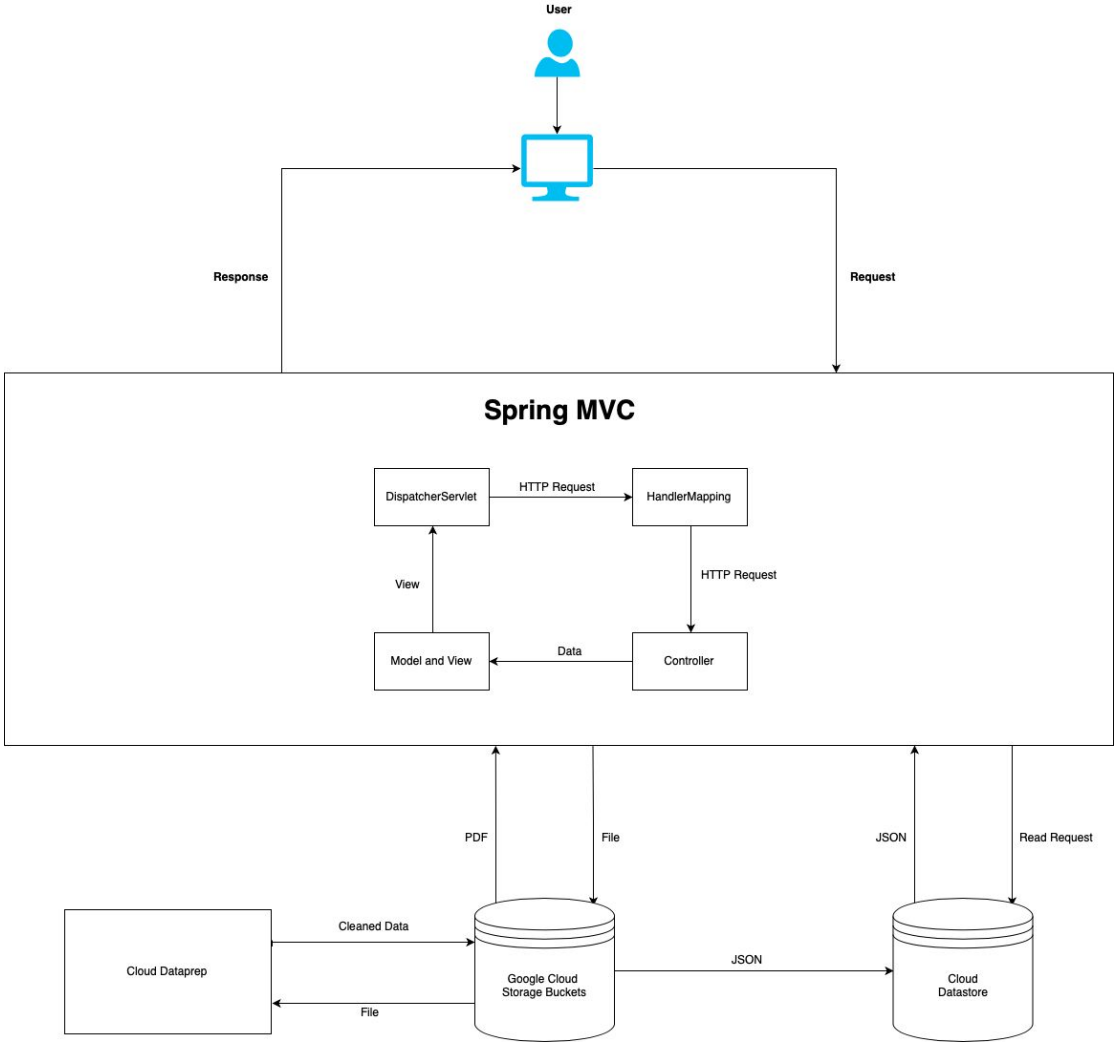
Description - The files uploaded by the user will be uploaded to buckets and the PDF reports will be stored in the users individual bucket
Criticality - Without buckets, there would be no location that an object could be stored in after being uploaded, or place for an object to be saved when the user is finished.
Technical issues - Each object in the buckets must be associated with a certain user and returned based on the user who requested it.
Dependencies with other requirements - File uploads

4. System Architecture

4.1 Architecture Overview Diagram

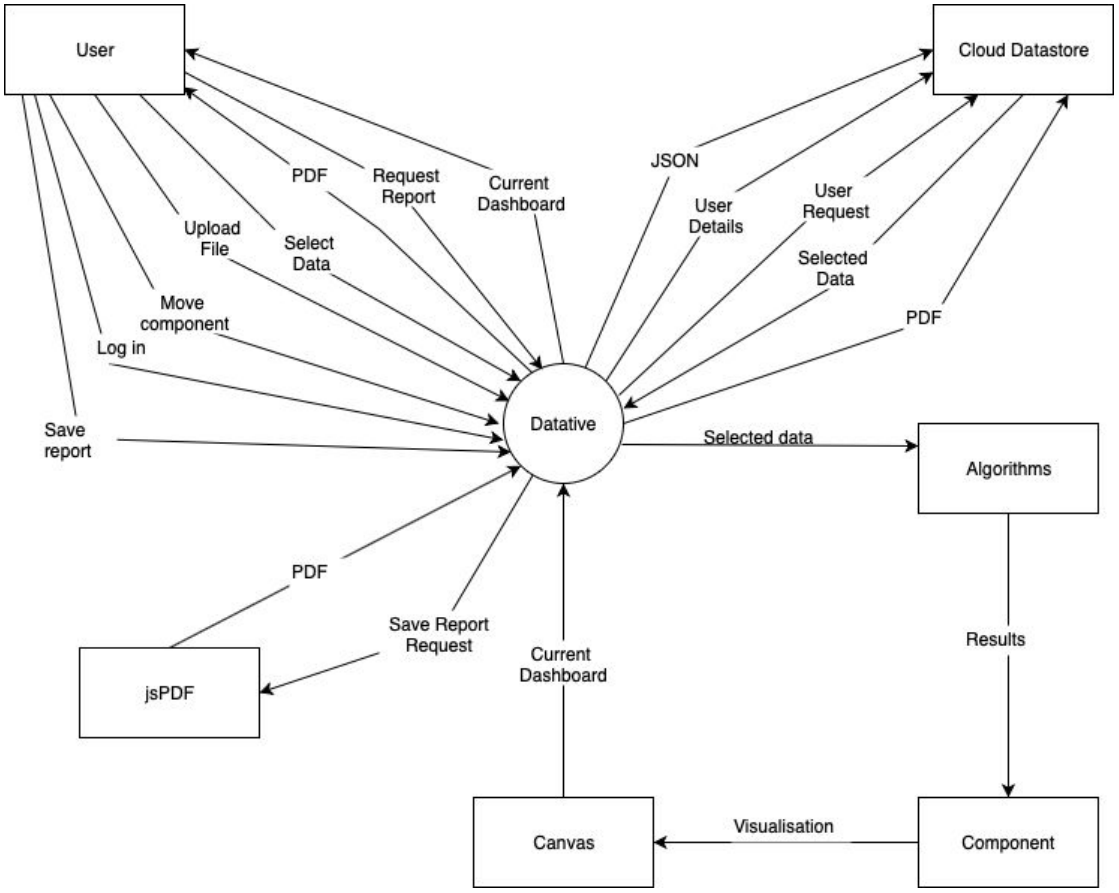


4.2 Application Architecture Diagram



5. High-Level Design

5.1 Context Diagram



5.2 Context Diagram Overview

User - The user entity is the person who can access the application after logging in

Cloud Datastore - Stores the data uploaded by the user. It can also return data requested by the user.

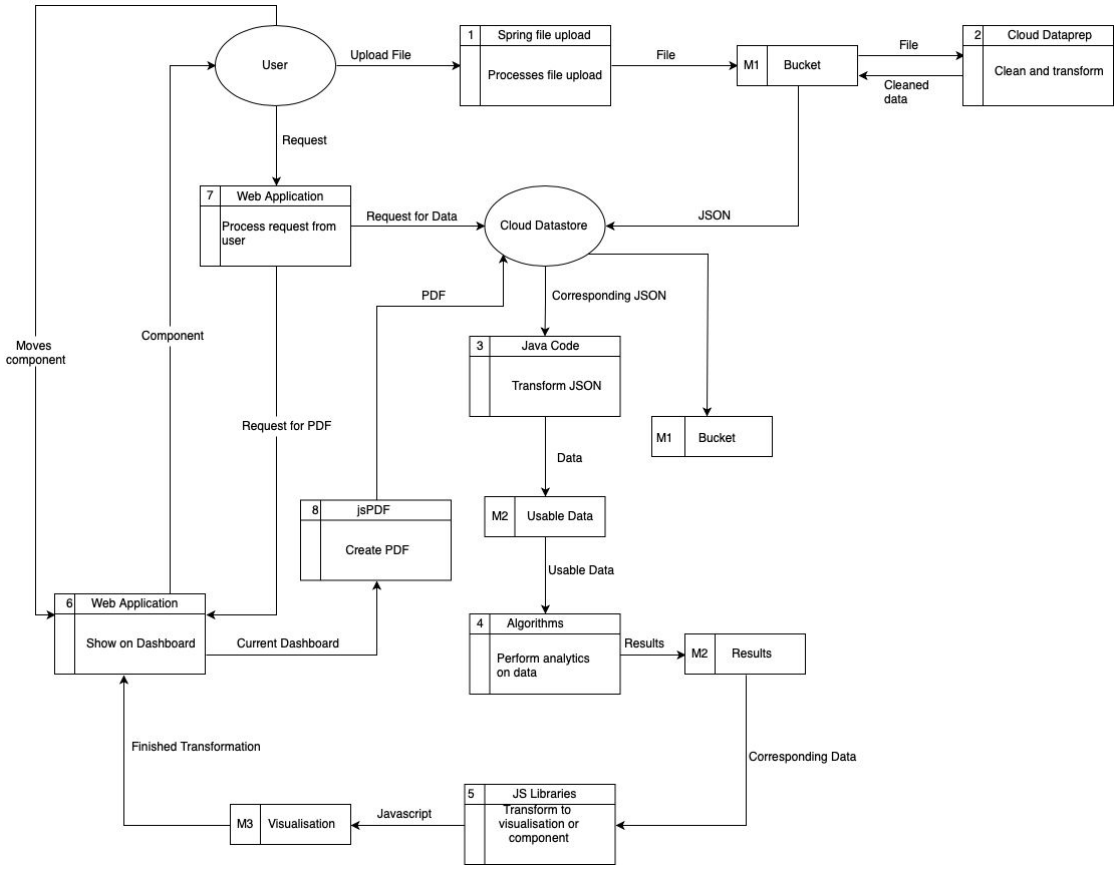
Algorithms - Algorithms which will be run against the data to create meaningful analytics

Component - A visual representation of the analytics generated by the application

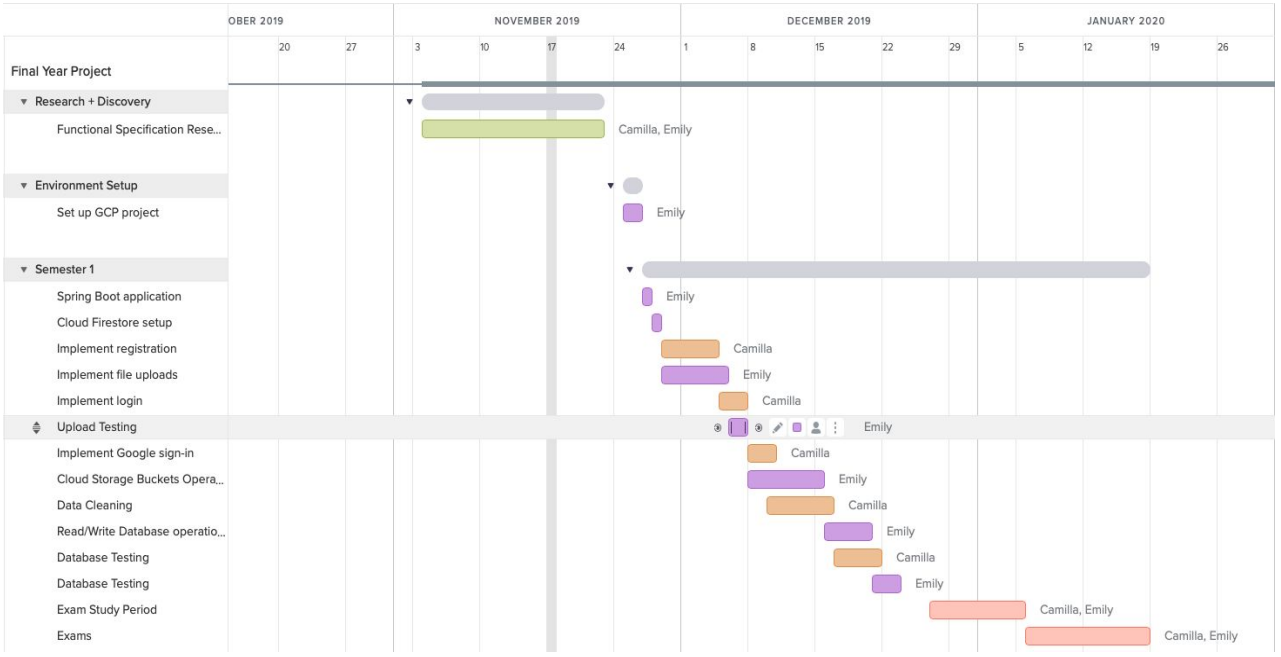
Canvas - The area in the application where components are put and moved around

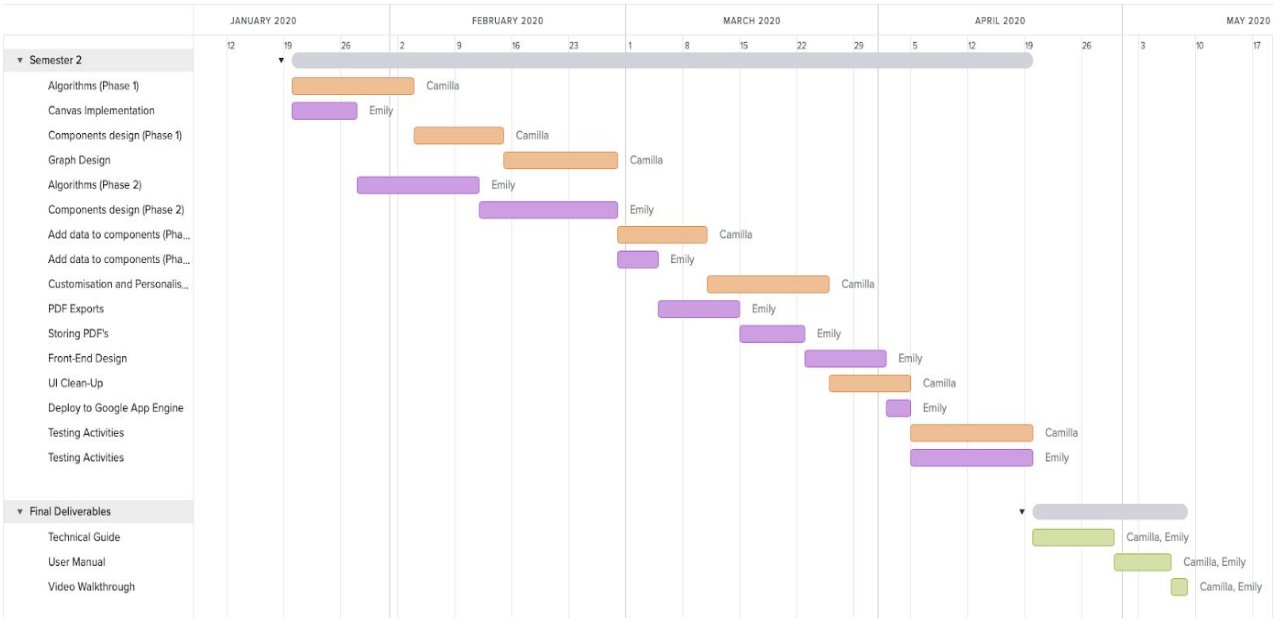
jsPDF - Generates a PDF of the dashboard designed by the user

5.3 Data Flow Diagram



6. Preliminary Schedule





7. Appendices

7.1 Resources

1. https://cloud.google.com/dataprep/
2. https://cloud.google.com/datastore/
3. https://github.com/MrRio/jsPDF
4. https://spring.io/projects/spring-boot
5. https://cloud.google.com/docs/
6. https://www.chartjs.org/docs/latest/
7. https://uber.github.io/react-vis/documentation/welcome-to-react-vis