

CS3219 – cir pROJECT

Group 4



LINK OF GIT REPOSITORY: https://github.com/MightyCupcakes/CS3219-Assignment-3

|  |  |  |
| --- | --- | --- |
| **Group MEMBERS** | | |
| **name** | **matric No.** | **Pair** |
| Chee Yi HSIEN SHAUN | a0139947l | 11 |
| GOH QI WEI | a0139046e |
| GOH YI DA JERemy | A0135768R | 12 |
| Tan boon JOON | A0139942W |

# Introduction

Our team was tasked to create an information retrieval application for users. As people understand and retain information better when it is visually presented, it is crucial that we communicate information in a quick and visual way. We developed an application with a user-friendly interface to help people visualise and analyse various aspects of research publications easily. This allows users to quickly interpret the data and adjust different variables to see their effects.

Allocation of work:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Work** | **Chee Yi Hsien Shaun** | **Goh Qi Wei** | **Goh Yi Da Jeremy** | **Tan Boon Joon** |
| Integration of code | Check mark symbol | Check mark symbol | Check mark symbol | Check mark symbol |
| Trend 1: Transition over time |  |  | Check mark symbol | Check mark symbol |
| Trend 2: Contemporary comparison |  |  | Check mark symbol | Check mark symbol |
| Trend 3: Top N X of Y |  |  | Check mark symbol | Check mark symbol |
| Trend 4: Network Citation (From base-paper) | Check mark symbol | Check mark symbol |  |  |
| Trend 5: Network Citation (Based on start and end year) | Check mark symbol | Check mark symbol |  |  |
| Web Server UI | Check mark symbol | Check mark symbol | Check mark symbol | Check mark symbol |
| Additional queries | Check mark symbol | Check mark symbol |  |  |
| Report and documentations | Check mark symbol | Check mark symbol | Check mark symbol | Check mark symbol |

# Requirement Specification

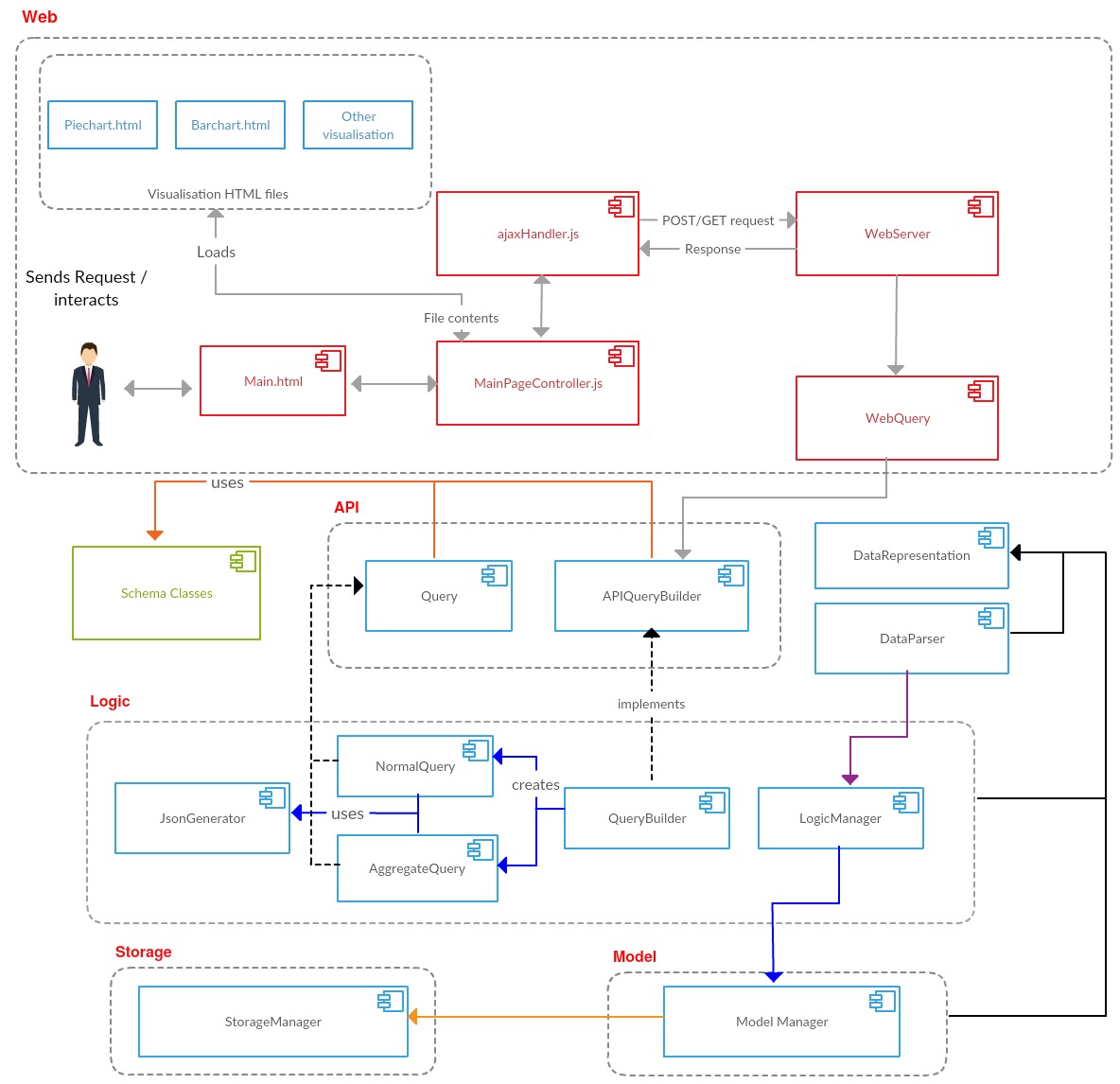
|  |
| --- |
| **Functional Requirements** |
| * The program should have a user-friendly interface * The program should create a visualization based on pre-made queries * The program should create a visualization based on customized queries by the user |

|  |  |
| --- | --- |
| **Non-Functional Requirements** | |
| **Usability** | * The program should notify the user when the visualization is created |
| **Reliability** | * The program should be able to handle exceptions and prompting the user for a new input |
| **Performance** | * TO BE DETERMINED |
| **Supportability** | * The program should be able to handle any future extensions * The program should be able to do unit testing on all individual components |
| **Scalability** | * The program should be able to handle large data sets |
| **Maintainability** | * The program should have excellent architecture design and good code quality |

# Design and Implementation

## Component Diagram (Need to re-draw)

Our team adopted a Page Controller(?) architecture as … . We decided to use Assignment 4 dataset.



|  |  |
| --- | --- |
| **Component** | **Description** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Design Decisions

We decided to use … because…

### Front-End (UI)

Our UI looks as follows:  
We chose it to be like this because …

### Back-End

We followed … architecture as … Maintainability, scalability and easy to expand etc..

## Visualizations

### Trend 1: Transition over time

|  |  |  |
| --- | --- | --- |
| **Objective** | **Visualization used** | **Purpose** |
| To find the trend of transition over time for a conference X |  | We used this visualization because |

\*insert visualization pic here\*

### Trend 2: Contemporary Comparison

|  |  |  |
| --- | --- | --- |
| **Objective** | **Visualization used** | **Purpose** |
| Compare different conferences over the same year |  | We used this visualization because |

\*insert visualization pic here\*

### Trend 3: Top N X of Y

|  |  |  |
| --- | --- | --- |
| **Objective** | **Visualization used** | **Purpose** |
| Where N is a number, X can be authors, conference citations, venue, booktitle, etc., and Y can be conference, author, cited authors, etc. |  | We used this visualization because |

\*insert visualization pic here\*

### Two queries of our choice

|  |  |  |
| --- | --- | --- |
| **Objective** | **Visualization used** | **Purpose** |
| Citation web for base paper with title X | Collapsible Tree | We decided to use a collapsible tree as … |

|  |  |
| --- | --- |
| **Step** | **Procedure** |
| 1 |  |
| 2 |  |
| 3 |  |

The picture below shows the visualisation for the base paper “gf(q)”.

\*insert visualization pic here\*

|  |  |  |
| --- | --- | --- |
| **Objective** | **Visualization used** | **Purpose** |
| Citation web for a selected base-paper specified by a user within a range of years | Forest |  |

|  |  |
| --- | --- |
| **Step** | **Procedure** |
| 1 |  |
| 2 |  |
| 3 |  |

\*insert visualization pic here\*

## API Development

Our team used the following API:

# Coding standards

Our team used Gradle (An open source build automation system) for our project. We used Eclipse to code our project and our version control was Github.  
  
We followed the Java syntax and naming conventions as we write our program. The details of the coding recommendations can be found at: <https://www.securecoding.cert.org/confluence/display/java/Java+Coding+Guidelines>  
  
To assist other developers in understanding how our code works, we have also written meaningful comments for our functions throughout the coding process.

# Testing

It is the responsibility of the developers to do unit testing. To practice good habits, our team has incorporated unit testing into our code. Some examples of our unit tests are as follows:

## 5.1 Example test case #1 (Logic)

|  |  |
| --- | --- |
| **Test Purpose** | Checks if we are able to get data with no citation |
| **Required Test Inputs** | TEST\_FOLDER that holds the parsed and raw data |
| **Expected Test Results** | Data with no citation |

## 5.2 Example test case #2 (File request from Web Server)

|  |  |
| --- | --- |
| **Test Purpose** | Checks if file request is handled properly |
| **Required Test Inputs** | WebServerHandler, URI |
| **Expected Test Results** | outputStream is the same as the file created |

# Additional Information (Optional)