

Table of Contents

Code Repository URL	4
Introduction	4
Requirements for the Existing Application	4
Existing Design	7
Overall Architecture	7
Sequence Diagram	8
Backend Architecture	9
Frontend Architecture	10
Evolution Requirements	10
Developer Documentation	19
Development Process	19
Agile Development	19
Continuous Development (CD)	19
Issue Tracker	19
Design Decisions	19
Redesign of the Architecture	19
Vue, Vuex and Spring Boot	20
Google App Engine	20
Google Cloud SQL	20
Design Diagrams	21
Overall Architecture	21
Sequence Diagram	22
Backend Architecture	23
Frontend Architecture	28
Code Implementation	34
Getting Started	34
Get a Copy	34
Environment Setup	34
Test and Deploy	35
Test	35
Deploy	35
Suggestions for Future Improvements and Enhancements	35
Project Based Dataset	35
More Detailed User Guide	36
Merging Multiple Visualizations into one Graph	36
Adding Dropdown Options to the Filter Function	36

Visualization Screenshots	36
Submission Record Charts	36
Word Cloud for Accepted Submissions Keywords	36
Word Cloud for All Full Papers Submissions Keywords	37
Word Cloud for All Submissions	37
Submission Rank Paper Author	38
Submission Rank Paper Author in Full Papers	38
Submission Acceptance Rate Rank Paper Author	39
Submission Accepted Rank Paper Author	39
Submission Rank Track	40
Acceptance Ratio Track	40
Acceptance Ratio by Year	41
Review Record Charts	41
Word Cloud for Reviewer Comment	41
Review Weighted Score Distribution	42
Review Weighted Evaluation Score Statistic Summary	42
Reviewer Expertise Level Statistic Summary	43
Reviewer Confidence Level Statistic Summary	43
Recommendation for Best Paper Distribution	44
Review Count Summary for Each Submission	44
Reviewer Assignment Rank	45
Reviewer Average Expertise Level Rank	45
Reviewer Average Confidence Level Rank	46
Reviewer Average Evaluation Score Rank	46
Number of Review Distribution	47
Reviewer Average Expertise Level Rank	47
Reviewer Average Confidence Level Rank	48
Reviewer Average Evaluation Score Distribution	48
Average Expert Level For Submission	49
Average Confidence Level For Submission	49
Author Record Charts	50
Submission Rank Author	50
Submission Rank Country	50
Submission Rank Organization	51
Submission + Author Record Charts	51
Submission Acceptance Rate Author Distribution	51
Submission Rank Author in Full Papers	52
Submission Acceptance Rate Rank Author	52
Submission Accepted Rank Author	53
Submission Acceptance Rate Rank Organization	53

Submission Accepted Rank Organization	54
Submission Acceptance Rate Rank Country	54
Submission Accepted Rank Country	55
Submission + Review Record Charts	55
Acceptance Rate and Weighted Score	55
Average Weighted Score By Track	56
Earliest Review in Days For Submission	56
Average Weighted Score Rank Paper Author	57
Author + Review Record Charts	57
Average Weighted Score Rank Author	57
Average Weighted Score Rank Organization	58
Average Weighted Score Rank Country	58

CS3219 Project Report

Code Repository URL

<https://github.com/CS3219-SEM1/chairvise-project-2018-team-01>

Student Name	Xiao Pu	Tan Heng Yeow	Yang Zhuohan	Li PengCheng
Matric Number	A0147978E	A0156105N	A0147995H	A0155563B

Introduction

Academic and industry conferences are held worldwide annually for researchers to present and discuss their work. Many leading conferences receive thousands of papers, which are reviewed by the Program Committee (PC) members, for their inclusion in the conference. The conference program Chair reports a summary of interesting facts e.g. acceptance and rejection rates, country specific contributions, during the conference based on the papers that are submitted.

These data could give insights on top upcoming research focus, geographical priorities and affinity to certain research areas which could be very useful to the researchers in the field. In addition, such insights could also help the organisers to make informed decisions to make the conference program more varied or more focused. With reference to the functional prototype that was provided to us, our team has built a brand new robust web application to enable conference program chairs to visualize and share their conference submission statistics.

Requirements for the Existing Application

The existing application consists of requirements that enable conference program chairs to visualize and share the conference submission statistics. We have grouped the existing requirements into user stories to illustrate their functions and benefits.

General requirements of the application		
User type/role	Function	Benefit
As a researcher	I can upload CSV files that need to be analyzed into a drag-and-drop box	So that I have a hassle-free file upload experience
As a researcher	I can choose to include or exclude certain sections in the exported PDF file	So that I can choose the sections I want to share with others
As a	I can get an auto-generated summary	So that I get to know the key

researcher	for some sections	findings presented in the section
------------	-------------------	-----------------------------------

Requirements for author.csv		
User type/role	Function	Benefit
As a researcher	I can view the top authors with a large number of submissions in bar chart	So that I can use this information to analyze the contribution level of authors
As a researcher	I can view the top countries with a large number of submissions in bar chart and pie chart	So that I can use this information to analyze the contribution level of countries
As a researcher	I can view the top universities with a large number of submissions in bar chart and pie chart	So that I can use this information to analyze the contribution level of universities

Requirements for review.csv		
User type/role	Function	Benefit
As a researcher	I can view the distribution of weighted average scores of all the papers submitted in bar chart	So that I have an idea of the general score of all the research papers reviewed
As a researcher	I can view the distribution of number of papers that are recommended for the best paper with scale from 0.0 to 1.0 in bar chart	So that I can know the quality of submission in terms of recommendation for the best paper
As a researcher	I can see the mean weighted evaluation score, average recommendation values and mean confidence level of all the reviews	So that I can learn more about the characteristics of the overall conference submissions

Requirements for submission.csv		
User type/role	Function	Benefit

As a researcher	I can view the association between submission time and last edit time series in line chart	So that I could know the relationship between those series and the relationship between those and the deadline
As a researcher	I can view the past year acceptance rate in line chart	So that I can analyze the trends of them
As a researcher	I can view the acceptance rate by track in bar chart or radar chart	So that I can compare the acceptance rate of them in different tracks.
As a researcher	I can view the top accepted author/contributors in bar chart	So that I can know the rank of authors based on the number of accepted papers.
As a researcher	I can view the top accepted author/contributor in different tracks in bar chart	So that I can know the rank of authors based on the number of accepted papers in different tracks.
As a researcher	I can view the word cloud for all submitted papers	So that I can know which topics are popular among all the submissions.
As a researcher	I can view the word cloud for all accepted papers	So that I can know which topics are popular among submissions which are accepted.
As a researcher	I can view the word cloud for submissions by tracks	So that I can know which topics are popular among different tracks.

Existing Design

Overall Architecture

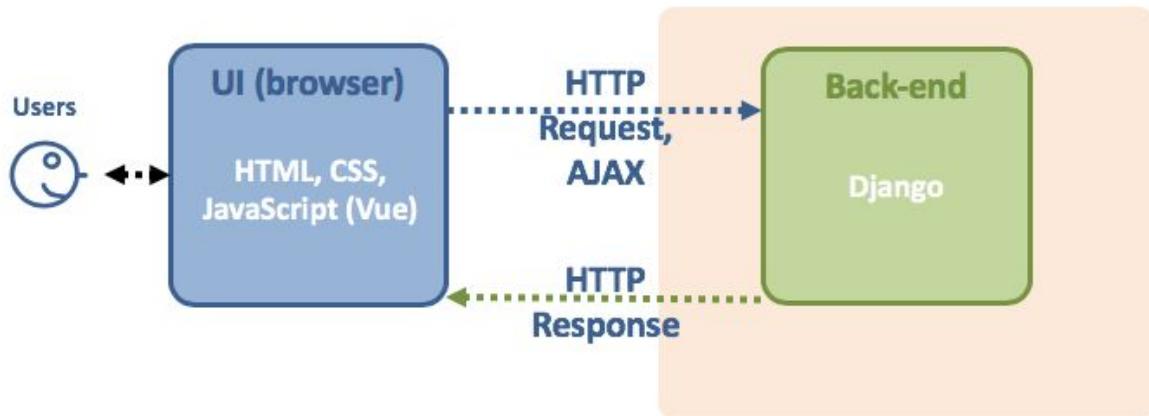


Figure 1

The overview of the main components is described below:

- **UI:** The User Interface (UI) seen by users consists of web pages containing Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript. JavaScript files are generated using Vue¹ through direct rendering of data straight to Document Object Model (DOM).
- **Back-end:** The back-end system is based on the Django² framework. It contains logic to group, aggregate and analysis data. Insights will be generated in JSON format and send to the frontend.

¹ Vue is modern front-end framework. <https://vuejs.org/>

² Django is a framework written in Python to build web application. <https://www.djangoproject.com/>

Sequence Diagram

The only Application Programming Interface (API) endpoint in the backend is to generate insights based on the uploaded CSV file. The overall sequence diagram of this action is shown as below.

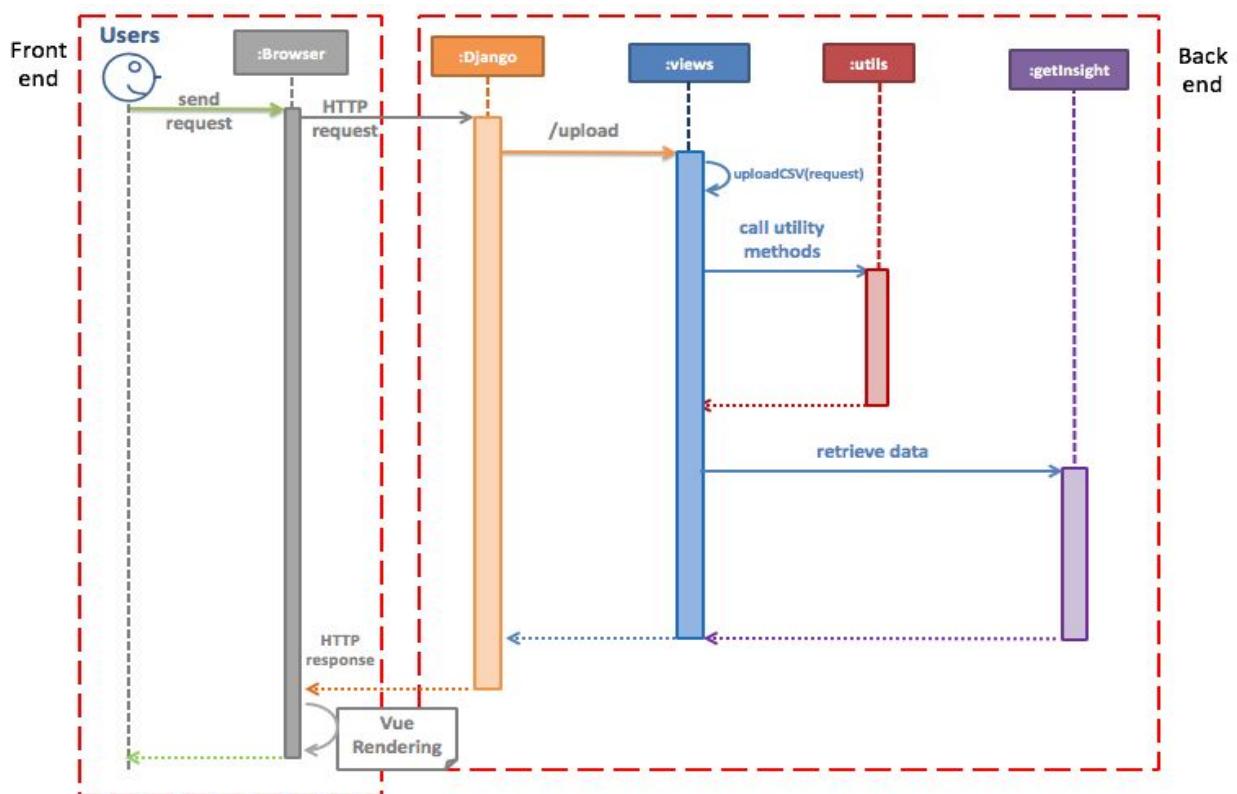


Figure 2

1. User sends a request to the browser to upload a CSV file.
2. Browser sends a HTTP request with the uploaded CSV file to the backend server.
3. The `/upload` endpoint is being accessed, which is directed to the function `uploadCSV()` in `views.py`.
4. Utility functions are accessed through `utils.py`.
 - a. First, it tests if the CSV file format matches with the desired ones
 - b. If the file format matches, it will parse the CSV file into a 2D list
5. Data is retrieved through `getInsight.py`. Depending on what the type of CSV is, it parses the 2D list into the respective insights.
6. CSV is parsed and data is returned as a JSON object as the HTTP Response
7. HTTP response is parsed in the front-end through Vue rendering.

Backend Architecture

The diagram below showcases the backend architecture overview of the existing system.

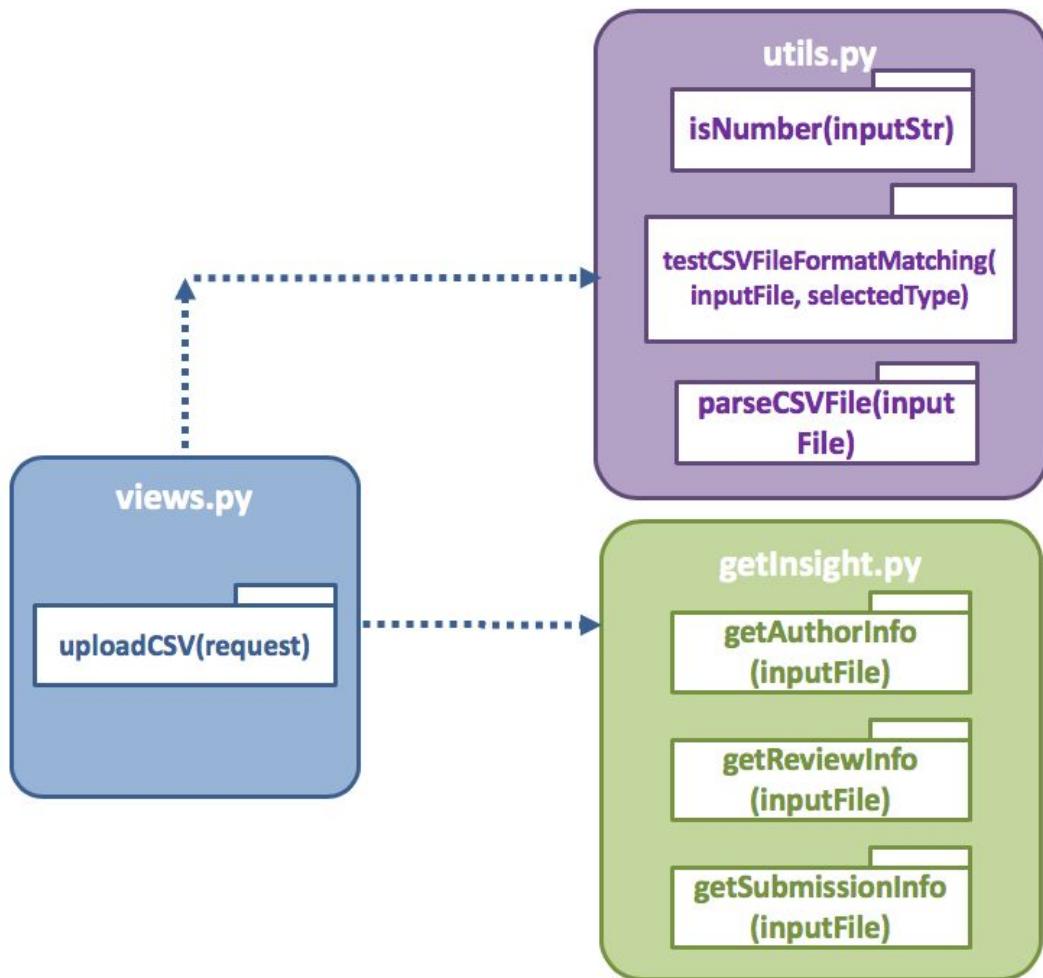


Figure 3

views.py

1. `uploadCSV(request)`: Handles HTTP request that access the `/upload` endpoint. Returns a HTTP response (JSON object) with 2 attributes ('`infoType`' and '`infoData`')

utils.py

1. `isNumber(inputStr)`: Tests if input string can be cast to a number
2. `testCSVFileFormatMatching(inputFile, selectedType)`: Tests if the CSV file format matches with the desired ones
3. `parseCSVFile(inputFile)`: Parse the CSV file and returns a 2D list

getInsight.py

1. `getAuthorInfo(inputFile)`: Parse the `author.csv` file and generate insights
2. `getReviewInfo(inputFile)`: Parse the `review.csv` file and generate insights

- getSubmissionInfo(inputFile): Parse the submission.csv file and generate insights

Frontend Architecture

The diagram below showcases the frontend architecture of the existing system.

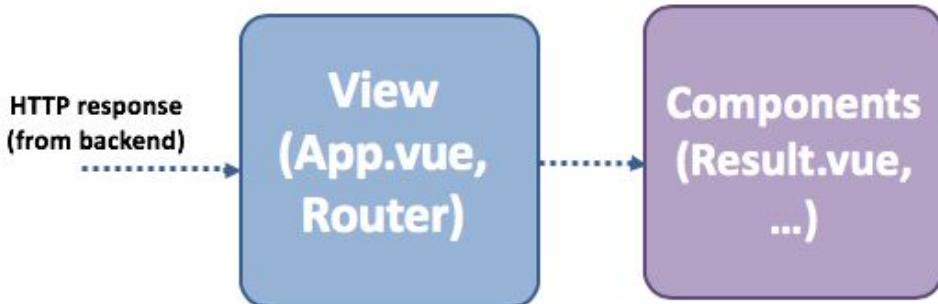


Figure 4

View

- App.vue**: The starting point of the application. This is where the root Vue object attaches to the DOM.
- The router provides 3 different router views (subpages) associated with it. They are home ('/'), TE ('/te'), and result ('/result'). This is used to achieve the goal of Single Page Applications (SPA).

Components

- Components allow reusability of display logic. **Result.vue** is the key file that will render the HTTP response and display the respective results to the user. **Result.vue** will also calculate properties and pass them to different components to display different charts.

Evolution Requirements

We have decided to keep selected requirements from the existing application that we feel would be useful for researchers. These requirements would be incorporated in our new system. In addition to the selected requirements, we have included new requirements that will enhance the experience of researchers.

Below is a list of new requirements that our team has decided to implement in the new version of our application. We are focusing on the following six categories:

- Making existing visualizations more meaningful for the user
- Adding more single visualisations that provide more value to the user
- Improve the user interface, making the system more useful and appealing
- Creating visualizations by joining multiple csv files
- Session management
- Mapping data schemes

Furthermore, we have decided to focus on three quality attributes for this project:

- Usability**: We have improved the user interface to make it easy for researchers to use the web application. This includes a comprehensive user guide as well as intuitive buttons for researchers to navigate around the application. In addition, we have

provided the user with more features to increase their user experience which would be shown in the refined requirements.

2. Modifiability: One of the many reasons why we built a new web application from ground up is to allow developer to adapt and add new visualizations easily. This is explained more in our design decisions.
3. Reusability: Our team also focuses on reusability to minimize the number of components required to be written (e.g. each component is specific to a diagram rather than a visualization). As such, developers can expect to reuse components without spending time to create new components for each potential new visualization needed in the future. This is explained more in our design decisions.

Making existing visualizations more meaningful for the user	
Previous visualization	Improved visualization
<p>Use Case: Show Bar/Line/Pie Chart Actors: Researcher</p> <ol style="list-style-type: none"> 1. Researcher visits page that contains bar/line/pie chart 2. System draws the bar/line/pie charts are draw 3. Researcher hovers each bar/line/pie to see the values associated with that bar 	<p>Use Case: Show Bar/Line/Pie Chart Actors: Researcher</p> <ol style="list-style-type: none"> 1. Researcher visits page that contains bar/line/pie chart 2. System draws the bar/line/pie charts are draw 3. System shows values associated with each bar/line/pie are draw together with the bar
<p>Use Case: Show top accepted authors/contributors Actors: Researcher</p> <ol style="list-style-type: none"> 1. Researcher visits page that contains the visualization 2. System splits the authors field by using ‘and’ delimiter 3. System counts number of accepted papers for each authors/contributors 4. System draws bar chart 	<p>Use Case: Show top accepted authors/contributors Actors: Researcher</p> <ol style="list-style-type: none"> 1. Researcher visits page that contains the visualization 2. Systems splits the authors field by using ‘and’ and ‘,’ delimiter 3. System counts number of accepted papers for each authors/contributors 4. System draws bar chart
<p>Use Case: Choose number of results to display Actors: Researcher</p> <ol style="list-style-type: none"> 1. System gives options 3, 5 and 10 2. Researchers choose one number from the given option for number of results to display 	<p>Use Case: Choose number of results to display Actors: Researcher</p> <ol style="list-style-type: none"> 1. System gives options from 1 to 30 2. Researchers choose one number from the given range for number of results to display
<p>As a researcher, I can view the distribution of number of papers are being recommended for the best paper with scale</p>	<p>As a researcher, I can view how many papers are being recommended for the best paper in a pie chart, so that I can know the</p>

<p>from 0.0 to 1.0 in bar chart, so that I can know the quality of submission in terms of recommendation for the best paper</p>	<p>quality of submission in terms of recommendation for the best paper.</p>
<p>Use Case: Show submission rank for country/organization in pie chart Actors: Researcher</p> <ol style="list-style-type: none"> 1. Researcher visits page that contains the visualization 2. Researcher chooses number of results to display X 3. System calculates number of submission for the top X countries 4. System draws the pie chart 	<p>Use Case: Show submission rank for country/organization in pie chart Actors: Researcher</p> <ol style="list-style-type: none"> 1. Researcher visits page that contains the visualization 2. Researcher chooses number of results to display X 3. System calculates number of submission for the top X countries. 4. System generates the new category 'Other' to contains number of submission from other country/organization 5. System draws the pie chart
<p>Use Case: Show rank of submission Actors: Researcher</p> <ol style="list-style-type: none"> 1. Researcher visits page that contains the visualization 2. Researcher chooses number of results to display X 3. System calculates number of submission for the top X author by using name to determine the uniqueness of a submission 4. System draws the bar chart 	<p>Use Case: Show rank of submission Actors: Researcher</p> <ol style="list-style-type: none"> 1. Researcher visits page that contains the visualization 2. Researcher chooses number of results to display X 3. System calculates number of submission for the top X author by using name and email to determine the uniqueness of a submission 4. System draws the bar chart
<p>Use Case: Calculate mean evaluation score and mean confidence level Actors: Researcher</p> <ol style="list-style-type: none"> 1. Researcher visits page that contains the statistics 2. System takes the average of all overall evaluation scores in user submitted data 3. System takes the average of all confidence level in user submitted data 4. System display two means 	<p>Use Case: Calculate mean evaluation score and mean confidence level Actors: Researcher</p> <ol style="list-style-type: none"> 1. Researcher visits page that contains the statistics 2. System calculates the weighted evaluation score for each submission 3. System calculates the average of those evaluation score 4. System calculates the average confidence level for each submission 5. System calculates the average of those confidence levels 6. System display two means

Adding more single visualisations that provide more value to the user			
Record type	User type/role	Function	Benefit
Submission	As a researcher	I want to know the submission rank of papers by authors in bar chart in descending order	So that I know which authors submitted more papers than others
	As a researcher	I want to know the number of papers submitted in each track	So that I know which track is more popular than other tracks
Review	As a researcher	I want to see a word cloud showing a list of keywords under reviewer's comments for reviewed papers.	So that I can analyze which keywords are commonly used in the reviewer's comments of reviewed papers
	As a researcher	I want to see a statistic summary of the reviewer expertise level based on the minimum, maximum value, the average, median score and the standard deviation of the weighted evaluation scores.	So that I can get an insight on how specialized each reviewer is in their review
	As a researcher	I want to see a statistic summary of the reviewer confidence level based on the minimum, maximum value, the average, median score and the standard deviation of the weighted evaluation scores.	So that I can get an insight on how confident each reviewer is in their review
	As a researcher	I want to know the review count summary showing the statistics of the number of reviews for each submission based on the minimum, maximum value, the average and median score for each submission	So that I can get an insight on how many reviews are provided for each submission
	As a researcher	I want to know the reviewer assignment rank through a bar chart with the the number of fields each reviewer is assigned to in descending order	So that I can know how knowledgeable each author is
	As a	I want to know the average	So that I can know how

	researcher	expertise level of each reviewer in descending order	skilled each reviewer is during the review
	As a researcher	I want to know the average confidence level of each reviewer in descending order	So that I can know how confident each reviewer is during the review
	As a researcher	I want to know the average evaluation score of each reviewer in descending order	So that I can get an insight on how generous the reviewer grade other papers in general
	As a researcher	I want to know the distribution of the number of reviews for each submitted paper	So that I have an insight on how many reviews are made for each submission
	As a researcher	I want to know the average expertise level distribution of reviewers	So that I have an insight on how specialized each reviewer is in their review
	As a researcher	I want to know the average confidence level distribution of reviewers	So that I have an insight on how confident each reviewer is in their review
	As a researcher	I want to know the average evaluation score distribution of reviewers	So that I have an insight on the evaluation score given by each reviewer
	As a researcher	I want to know the average expert level for each submission	So that I have an insight on the average expertise level for different number of submissions
	As a researcher	I want to know the average confidence level for each submission	So that I can get an insight on how confident, from a range of 1 to 5, the reviewers are in reviewing all the submitted papers

Improve the user interface, making the system more useful and appealing
UI specification: The application should have a proper favicon and logo
UI specification: There should be a user guide presented when user visit the application

UI specification: There should be a menu bar navigate the user guide
UI specification: There should be a menu bar at top of the application for user to navigate
UI specification: There should be an loading icon showing when the system is processing user requests
UI specification: There should not be any internal system related information (e.g. framework user guide, framework generated sample page) exposed to users
UI specification: All UI element should used Element UI ³ components to ensure the consistent displaying
User Story: As a researcher, I can filter undesired data out in generating unrelated results from new visualizations, so that I can focus on the results that are important to me in the generated visualizations
User Story: As a researcher, I can preview the visualization when I filter out undesired data, so that I can review the changes before making the final change
User Story: As a researcher, I can associate extra data with each bar in bar chart, so that I can get more information about the group corresponds to the bar

Creating visualizations by joining multiple csv files			
User type/role	Group	Function	Benefit
As a researcher	author.csv and submission.csv	I want to know the percentage of acceptance rate of each author's papers in descending order	So that I can know which authors have higher acceptance rate than other authors
As a researcher		I want to know the number of accepted papers submitted by each author in descending order	So that I can know which authors has more accepted papers than others
As a researcher		I want to know the percentage of acceptance rate of each organization's papers in descending order	So that I can know which organizations has higher acceptance rate than other organizations
As a researcher		I want to know the number of accepted papers submitted by each organization in	So that I can know which organizations has more accepted papers than

³ Element UI is UI framework for Vue. <http://element.eleme.io/#/>

		descending order	other organizations
As a researcher		I want to know the percentage of acceptance rate of each country's papers in descending order	So that I can know which countries have higher acceptance rate than other countries
As a researcher		I want to know the number of accepted papers submitted by each country in descending order	So that I can know which countries have more accepted papers than other countries
As a researcher		I want to know the top accepted papers in the "Full Papers" track.	So that I know the ranking of authors in each individual track
As a researcher		I want to know the distribution of submission acceptance rate over authors	So that I would know whether the acceptance guideline is reasonable
As a researcher	review.csv and submission.csv	I want the percentage of acceptance rate based on the weight score of submissions	So that I can get an insight on what percentage of papers will be accepted given a review score
As a researcher		I want to see the number of days taken for submissions to be reviewed	So that I can get an insight on the efficiency of reviewers
As a researcher		I want to see the average weighted score of papers in each track	So that I can get an insight on which track has a higher weighted score than the other tracks
As a researcher		I want to see the average weighted score of all submissions of each author	So that I can get an insight on which author performs better
As a researcher	review.csv and author.csv	I want to see the average weighted scores of all submissions of each author	So that I can get an insight on which author performs better with his or her submission compared to other authors
As a researcher		I want to see the average weighted scores of all submissions for each organization	So that I can get an insight on which organization performs

		organization	better compared to other organizations
As a researcher		I want to see the average weighted scores of all submissions for each country	So that I can get an insight on which country performs better compared to other countries

Session management		
User type/role	Function	Benefit
As a researcher	I can create a session with name and description	So that I can describe what the session is about
As a researcher	I can create a session with arbitrary visualization I chosen	So that I have more control on what I want to analysis
As a researcher	I can get the list of all sessions I created	So that I can easily find them to retrieve old sessions.
As a researcher	I can share the session to public with read/edit permission	So that I can share my findings to others with permission
As a researcher	I can share the session to a specific person with read/edit permission	So that I can share my findings to others with permission

Mapping data schemes		
User type/role	Function	Benefit
As a researcher	I can map column names of CSV files to a standard data template using an mapping tool	So that I can analyze data that does not follow the order or naming of the data schema provided in the system
As a researcher	I can choose predefined mapping for CSV files that the system knows	So that I do not need to map each column repeatedly when reimport data

Discarded requirements of previous application
--

Previous user story	Current user story	Decision
As a researcher I can get an auto-generated summary for some visualizations, so that I get to know the key findings presented in the visualization	As a researcher I can get an description of each visualization, so that I get to know the key findings presented in the visualization	The naive auto-generated summary only scratches the surface of visualizations. User can always read the description of each visualization and get the key findings from the chart.
As a researcher, I can view the top countries with a large number of submissions through a pie chart and bar chart, so that I can use this information to analyze the contribution level of countries	As a researcher, I can view the top countries with a large number of submissions through a pie chart, so that I can use this information to analyze the contribution level of countries	Bar chart does not provide much more meaningful information than pie chart
As a researcher, I can view the top organization with a large number of submissions through a pie chart and bar chart, so that I can use this information to analyze the contribution level of organization	As a researcher, I can view the top organization with a large number of submissions through a pie chart, so that I can use this information to analyze the contribution level of organization	Bar chart does not provide any more meaningful information than pie chart
As a researcher, I can see the mean weighted evaluation score, average recommendation values for all reviews and mean confidence level of all the papers, so that I can learn more about the characteristics of the overall conference submissions	As a researcher, I can see the mean weighted evaluation score and mean confidence level of all the papers, so that I can learn more about the characteristics of the overall conference submissions	The average recommendation values for all reviews are not useful as recommendation decision is a binary value ('yes' or 'no').
As a researcher, I can view the acceptance rate by track in bar chart or radar chart, so that I can compare the acceptance rate of them in different tracks.	As a researcher, I can view the acceptance rate by track in bar chart, so that I can compare the acceptance rate of them in different tracks.	Radar chart does not provide any more meaningful information than pie chart

Developer Documentation

Development Process

Agile Development

Our group has decided to adopt an Agile Development Model. At the start of the project, we came up with a list of enhancements that we hoped to see from the existing implementation. As such, we set different milestones to be achieved at different phases of the semester. After coming up with the decision to start from scratch, our code base was set up within the first week. After each week, we aimed to have incremental improvements such as enhancing existing visualization, adding new visualizations. These functionalities were developed hand in hand with data mapping functionality and access control. One iteration was about one week long. All of us strived to clear all the issues assigned in each iteration so that we can constantly work on new interaction.

Continuous Development (CD)

We believe that CD is part of the Agile development process. In the initial stages, we have set up Travis on Github to automatically build and test our software. We have added both tests for the backend and frontend functionalities. We focused heavily on backend functionalities as we want to ensure that our new code will not break the workflow of our software at any point of time. This helped us to streamline the process of building, testing and deploying new code efficiently in a live server. This minimized the time for our project to build, test and release new features while ensuring that our product is maintained. Along the way, we also constantly tested both our live and local server to ensure that there was no disparity between the two.

Issue Tracker

In our Github Repository, we started off with a list of issues which we aim to tackle. We allocated different priority levels and tags to each of them. We mainly focused on issues with priority levels 1 and 2. These were the ones which we hope to deliver while priority level 3 are meant to be for future developments. We also added labels such as enhancement, bug, duplicate to handle a myriad of issues which we encountered through the semester. We also added a first timer label, not exploited, to guide anyone who may be interested in this project in the future.

Design Decisions

Redesign of the Architecture

Our group aimed to finish all six improvements and enhancements proposed by the earlier development team. After a detailed investigation, we found that supporting join schema and addition of more visualizations are big challenges under the existing architecture. Indeed, in the existing application, we have found a lot of duplicated code on grouping, aggregation and formatting to generate insights. It was foreseen that complex code would be introduced when more and more visualization are added. Therefore, a dedicated framework should be developed to modularized the grouping, aggregation and formatting of data. After some research, it was found that there is already sophisticated solution in the industry to solve

online analytical processing (OLTP) problems. Thus, we chose a very light solution using SQL with MySQL to analysis data and generate insights. In doing this, the concerns of aggregating and grouping data were pushed to the MySQL query engine and we could focus more on devising more visualizations.

Vue, Vuex⁴ and Spring Boot⁵

The frontend existing application was written in Vue. Therefore, our team sticked to the Vue for the frontend framework as some existing visualization code can be reused. For backend, Spring Boot was chosen over Django as Java is the language that all of our team members are proficient with. There was nothing much we can reuse from the existing backend as the backend shall be rewritten because of the change of architecture.

Google App Engine

Google App Engine (GAE) abstracts the low-level details such as HTTP server implementation. It lets developer focus on high-level business logic and interaction and ease the process of deployment. Therefore, to simply the development process, GAE is chosen to be our server.

Google Cloud SQL

As mentioned above, MySQL is chosen to be the backend database which is responsible to store both application data and to analysis queries sent by users. Setting up a MySQL server might be complex and time consuming. Fortunately, Google Cloud SQL provides pre-configured MySQL instance. The database also works well together the Google App Engine.

⁴ Vuex is a state management pattern + library for Vue.js applications. <https://vuex.vuejs.org/>

⁵ Spring Boot is designed to develop something as quickly as possible, with minimal upfront configuration of Spring. <https://spring.io/>

Design Diagrams

Overall Architecture

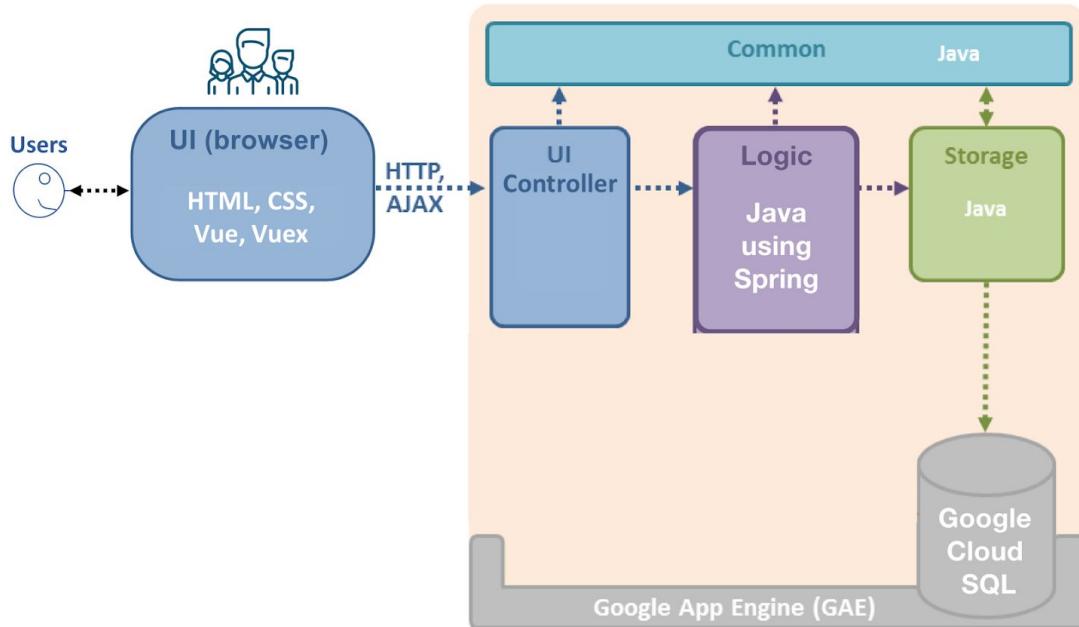


Figure 5

The application is designed with layered design pattern. Single Responsibility Principle (SRP) is applied here. Each layer/classe has its own responsibility.

- **UI**: The component consist of API controllers and WebPage controllers. API controllers are responsible to handle API calls by the frontend. WebPage controllers are responsible to serve static production Vue files.
- **Logic**: The main logic of the application is in Java using the Spring framework.
- **Storage**: The storage layer of the application uses the persistence framework provided by Google App Engine, using MySQL 5.6.
- **Common**: The Common component contains utility code (data transfer objects, helper classes, etc.) used across the application.

Sequence Diagram

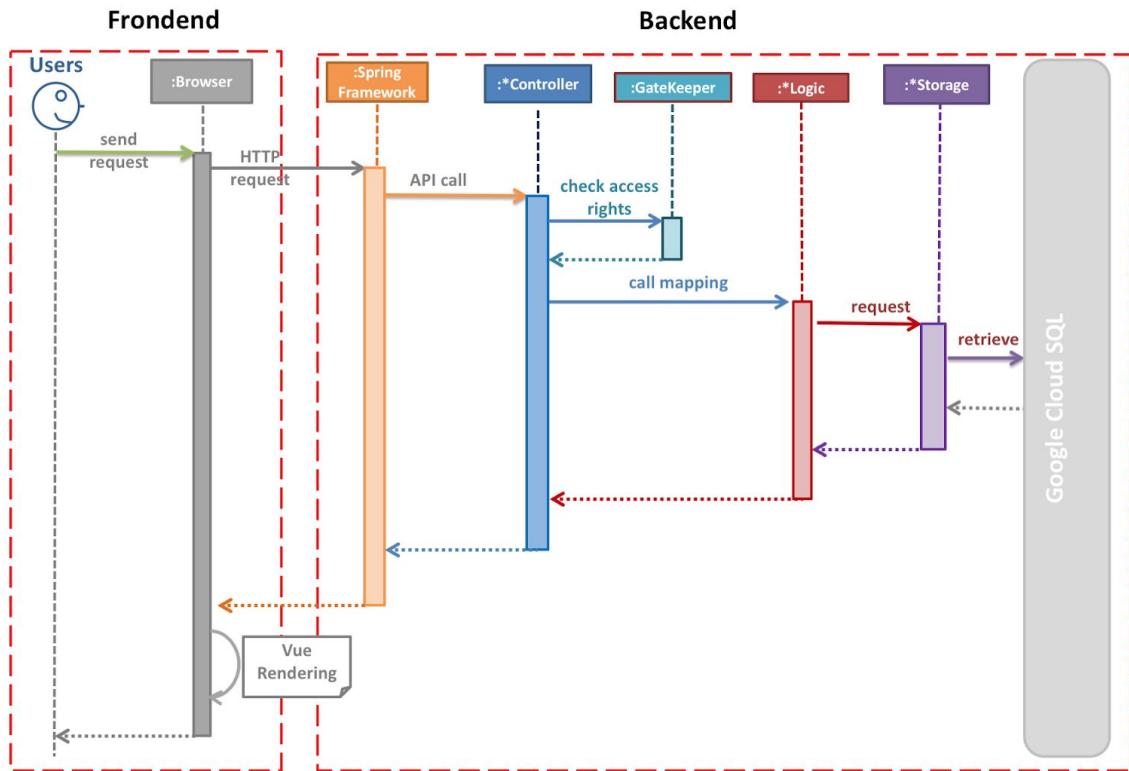


Figure 6

The diagram below showcases the workflow in the system of a typical request from the user.

1. User sends a request to the browser for certain data or visualisation.
2. Browser sends a HTTP request to the backend. Spring framework will handle the HTTP request and pass to the correct controller based on URL mapping.
3. Controller checks the access rights of the user and executes the call by interacting with the **GateKeeper**.
4. Logic processes and requests the data from the storage component.
5. Storage retrieves the data from Google Cloud SQL.
6. The response will be sent back to the browser where it will use Vue to render the web page.

Backend Architecture

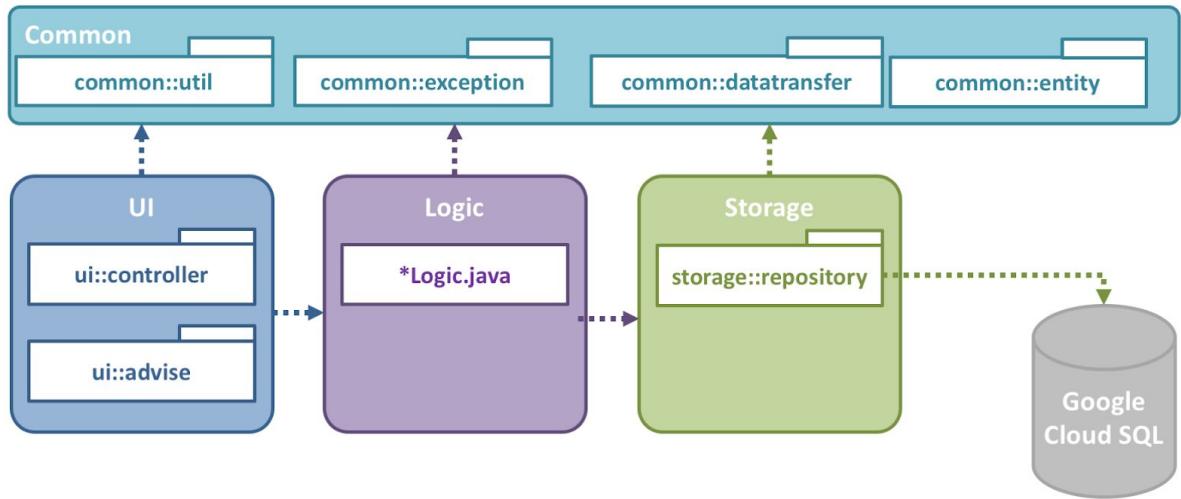


Figure 7

The diagram above illustrates the backend package overview of the existing system.

UI Component

The UI component is the first stop for all requests received by the backend of the web application.

- `ui.controller.api`: Provides backend Representational State Transfer (REST) API access to the users
- `ui.controller.data`: Contains helper objects to be sent to the client in JSON
- `ui.controller.webpage`: Handles static file requests for the users
- `ui.advice`: Handles exception thrown by the application

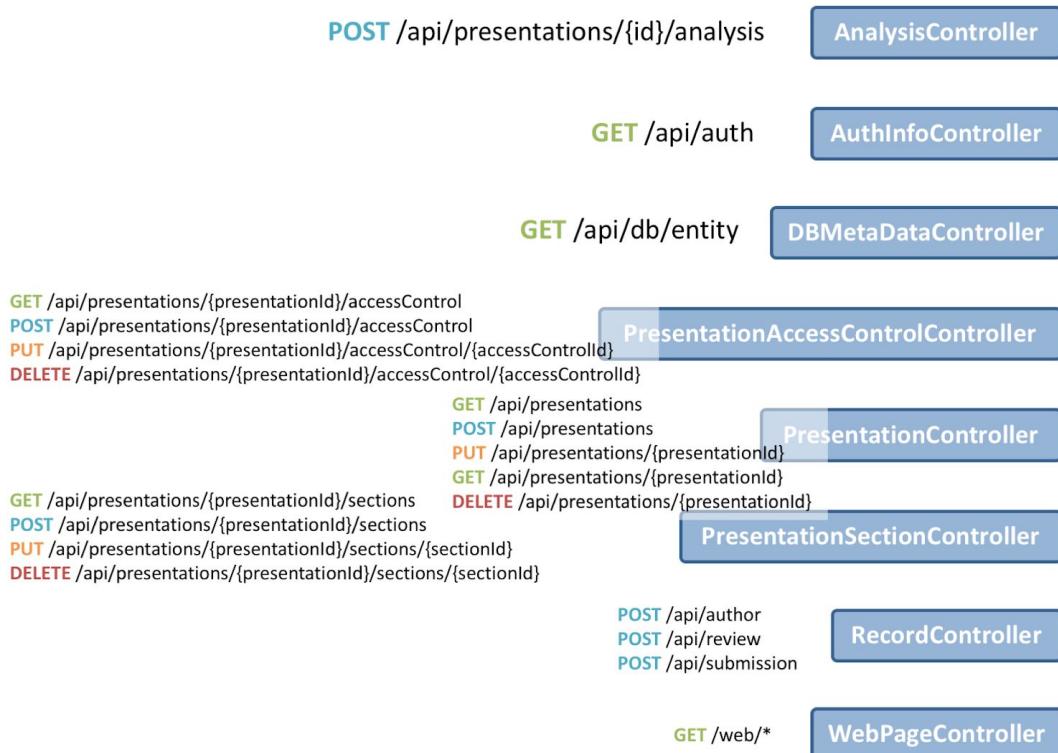


Figure 8

Figure 8 shows the mapping URLs and request method to the controller. In general, `GET` is used to retrieve data. `POST` is used to create data. `PUT` is used to update data. `DELETE` is used to delete data.

- **AnalysisController** is used to handle the analysis request send by the frontend and issue SQL aggregation query to the backend.
- **AuthInfoController** is used to check the current authentication status of user. For example, it will give login URL if user is not logged in and logout URL if user is logged in.
- **DBMetatDataController** exposes the metadata including name and type of the standard data template used in the application. This is useful when users try to match their own CSV file to the data template used in the application.
- **PresentationController**, **PresentationSectionController** and **PresentationAccessControlController** are responsible for the CRUD operations.
- **RecordController** accepts the CSV data imported by users and stores them in the database
- **WebPageController** serves the static production files built by Vue.

```

1  {
2    selections: [
3      expression: "ROUND(SUM(CASE WHEN s_is_accepted = 'accept' THEN 1 ELSE 0 END)/COUNT(*), 2)",
4      rename: 'acceptance_rate'
5    },
6    {
7      expression: "a_organisation",
8      rename: 'a_organisation'
9    },
10   {
11     expression: "COUNT(*)",
12     rename: 'submitted'
13   },
14   {
15     expression: "SUM(CASE WHEN s_is_accepted = 'accept' THEN 1 ELSE 0 END)",
16     rename: 'accepted'
17   },
18   involvedRecords: [
19     name: 'author_record',
20     customized: false,
21   },
22   {
23     name: 'submission_record',
24     customized: false,
25   },
26   ],
27   filters: [],
28   joiners: [
29     {
30       left: "a_submission_id",
31       right: "s_submission_id",
32     },
33   ],
34   groupers: [
35     {
36       field: "a_organisation"
37     }
38   ],
39   sorters: [
40     {
41       field: 'accepted',
42       order: 'DESC',
43     },
44     {
45       field: 'a_organisation',
46       order: 'ASC',
47     }
48   ]
49 }

```

Figure 9

Figure 9 shows a typical analysis request receive by `AnalysisController`. The request tries to rank organization based on the number of accepted submission that organization has. It contains `selections`, `involvedRecords`, `filters`, `joiners`, `groupers` and `sorters`, which are enough to generate an SQL query for aggregation. For instance, the above analysis request will generate the below SQL query (the generation logic is inside Logic Component).

```

SELECT Round(Sum(CASE
                      WHEN s_is_accepted = 'accept' THEN 1
                      ELSE 0
                  END) / Count(*), 2) AS `acceptance_rate`,
       a_organisation AS `a_organisation`,
       Count(*) AS `submitted`,
       Sum(CASE
                      WHEN s_is_accepted = 'accept' THEN 1
                      ELSE 0
                  END) AS `accepted`
FROM author_record,
     submission_record
WHERE author_record.data_set = 'user@email.com'
      AND submission_record.data_set = 'user@email.com'
      AND a_submission_id = s_submission_id
GROUP BY a_organisation
ORDER BY accepted DESC,
         a_organisation ASC

```

The query result will be sent to the frontend to display visualization.

It should be noted that under such data aggregation framework. Any new visualization can be easily supported.

Logic Component

The Logic component handles the business logic. In particular, it is responsible for:

- Managing CRUD operations, ensuring the integrity of data.
- Providing a mechanism for checking access control rights.

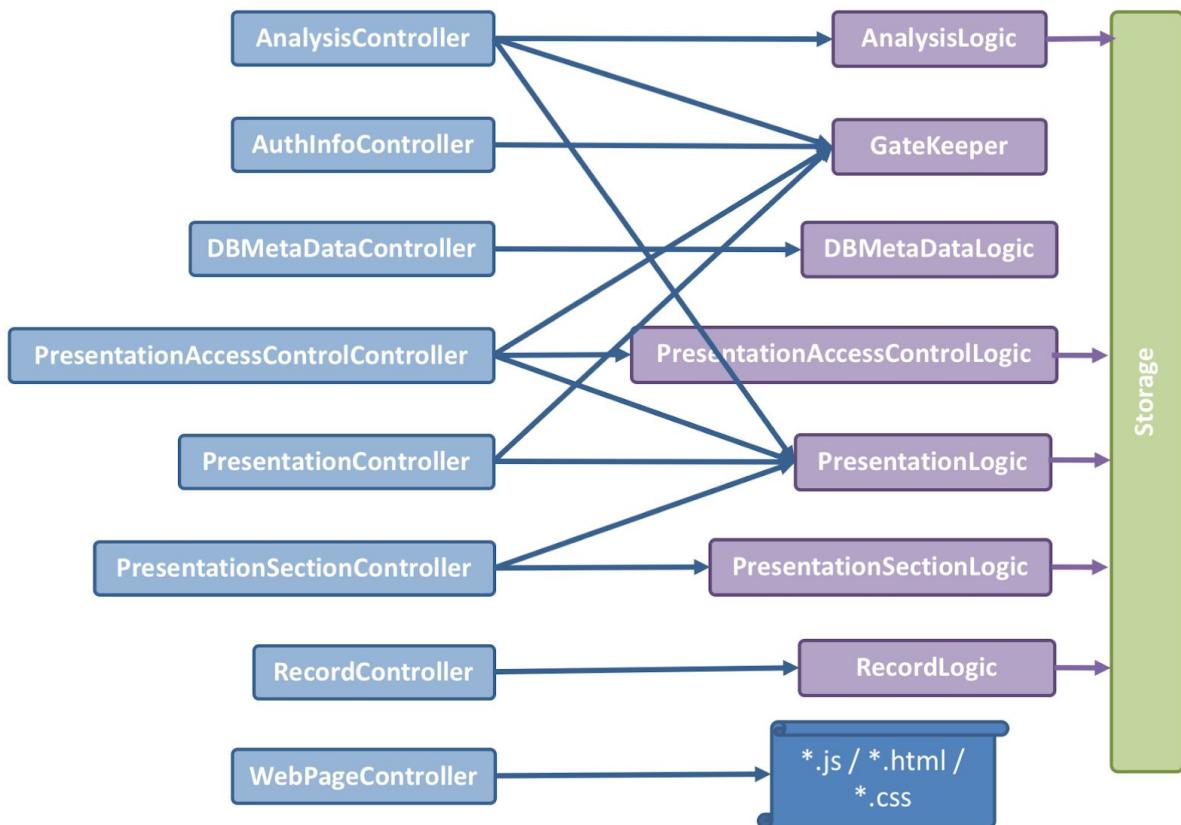


Figure 10

Figure 9 shows how each controller is making use of its corresponding logic. The dependency is injected (Dependency Injection) by Spring framework automatically.

Storage Component

The Storage component performs create, read, update and delete (CRUD) operations on data entities individually. It contains minimal logic beyond what is directly relevant to CRUD operations.

Figure 10 shows how logic makes used of the repository in the storage component.

- **AnalysisLogic** will issue SQL query directly to the Database to perform analysis query

- GateKeeper will use GAE internal APIs to check logged in user and also issue queries to database to check the access rights.
- *RecordRepository are responsible for persisting user uploaded CSV data to the database
- Presentation*Repository are responsible for storing user generated presentation, access control list and sections.

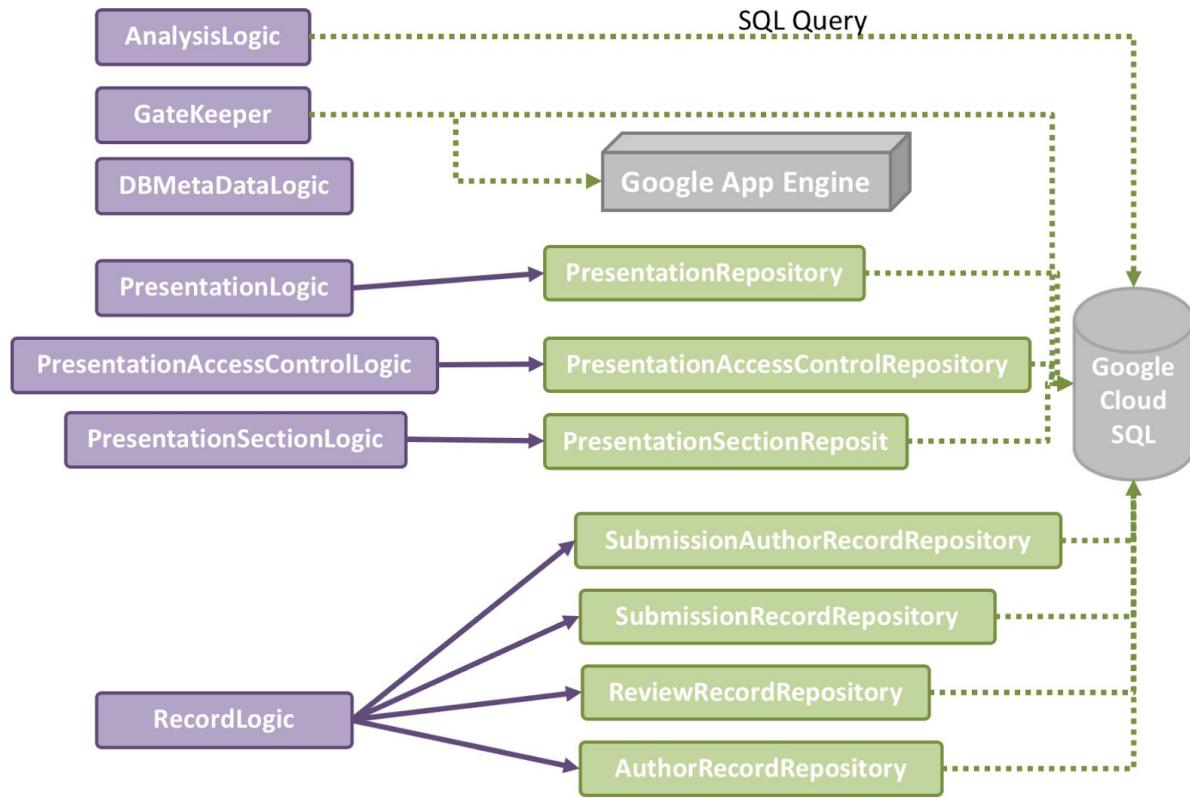


Figure 11

Common Component

The Common component basically contains common utilities used across the web application.

- **common.util**: Contains utility classes.
- **common.exceptions**: Contains custom exceptions.
- **common.datatransfer**: Contains data transfer objects (DTOs).
- **common.entity**: Contains entity stored in the database.

Figure 11 shows the entity relationship. Each entity has its corresponding table in the relational database.

- One presentation can have multiple sections or access controls.
- *Record is the data imported by users. There is a many-to-many relationship between **SubmissionRecord** and **SubmissionAuthorRecord** because one submission can be contributed by multiple authors. Each author can also contribute to multiple submissions.

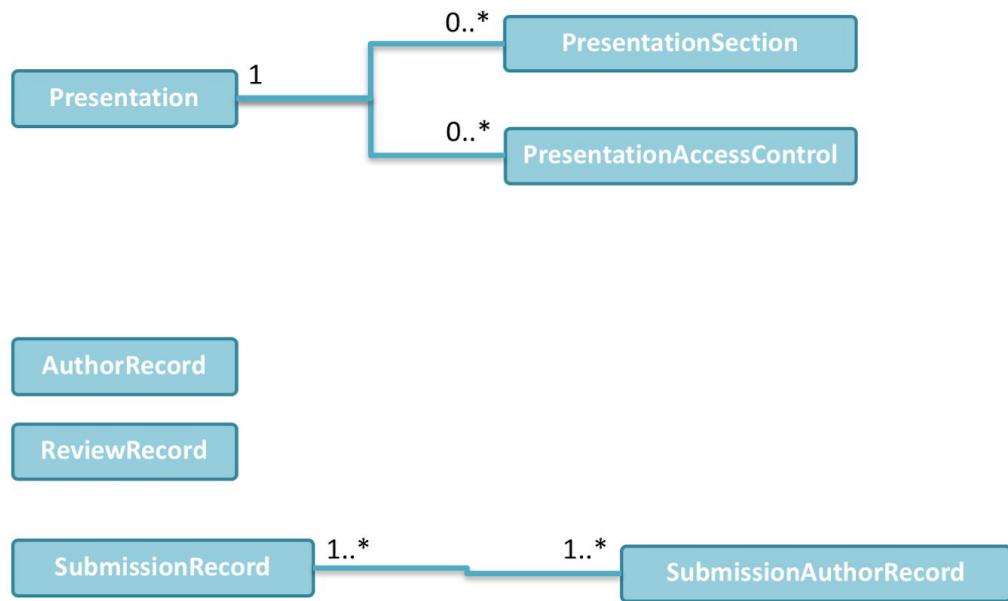


Figure 12

Frontend Architecture

Frontend Package

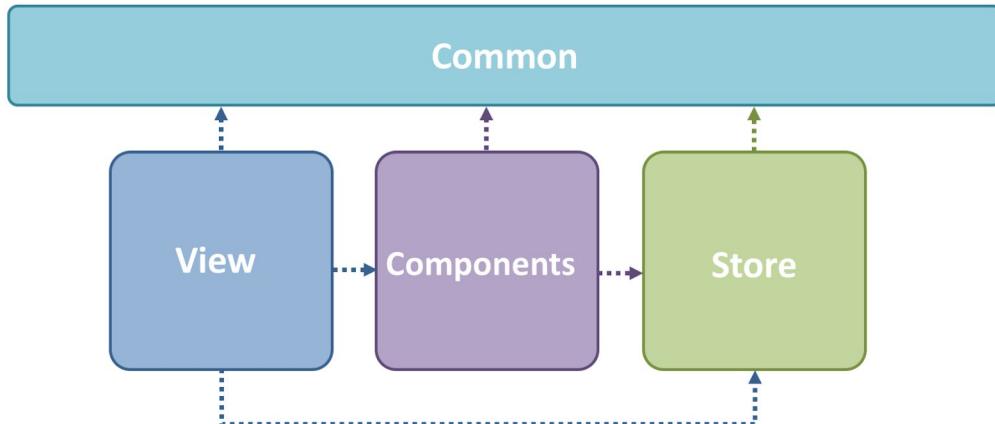


Figure 13

The diagram above illustrates the frontend package overview of the application.

- **view**: The view is mainly in charge of displaying pages of the application.
- **component**: The component contains reusable UI and display logic components which is called by multiple pages.

- **store**: The store contains core logic of the application. It shows the state of the application and handles mutation and action to interact within the application using Vuex.
- **common**: Contains utility function as well as constants used.

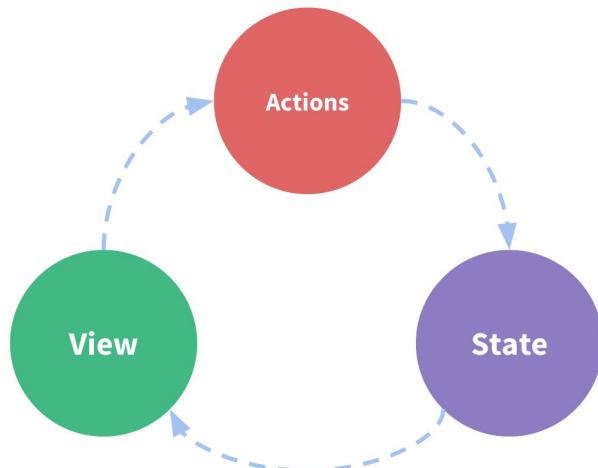


Figure 14. Source: Vuex Documentation

As shown in Figure 13. The view (components) will fire actions and the actions change the state. After that, as view is subscribed to the state (Observable Pattern), the UI (view) will be notified and updated. In doing this, the state becomes the single point of truth in the whole application, which increase the maintainability a lot and make it is easy to trace the change of UI.

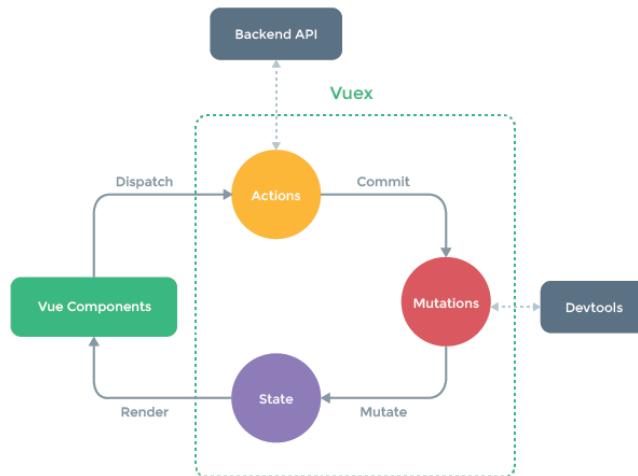


Figure 15. Source: Vuex Documentation

A more detailed diagram is shown in Figure 14. Typically, a vue component will dispatch an action. The action will communicate with backend API to do CRUD operations. After that, it will commit a mutation and the mutation will change the state. The change of state will be

propagated to the component and component will render based on the data in the state. A devtool (e.g. Vuex.js devtools⁶) can be used to monitor the state for easy debugging.

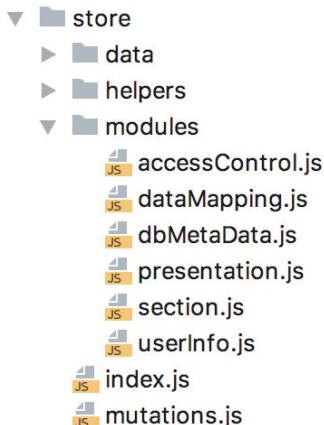


Figure 16

Indeed, in the store component, we have the following modules as shown in .

- `accessControl.js` is responsible for handling logic related to the access control of a presentation
- `dataMapping.js` is responsible for handling logic related to the mapping of CSV and built-in data schema
- `presentation.js` is responsible for handling logic related to the presentation
- `section.js` is responsible for handling logic related to the presentation section
- `userInfo.js` is responsible for handling logic related to the authentication

In summary, with the help of Vuex, the overall architecture of front end ensures that the state can only be mutated in a predictable fashion.

The components are designed to be highly reusable. The diagrams below break down the each page (view) into different components.

⁶ Vuex.js devtools is available as a Chrome Extension

<https://chrome.google.com/webstore/detail/vuejs-devtools/nhdogjmejiglipccpnnaanhbledajbpd?hl=en>

Home.vue

List of Information Available

ListOfGuide.vue

- General User Guide
- Features Supported
- How to Import Data
- How to Add Presentation
- How to Share Presentation

WelcomeMessage.vue



Welcome to Conference Data Visualization Management System

This project is designed to enable conference program chairpersons to visualize and share conference submission statistics. By parsing information in different formats, we aim to assist user to obtain the most valued out of the information uploaded. We also support sharing and exporting of such visualization.

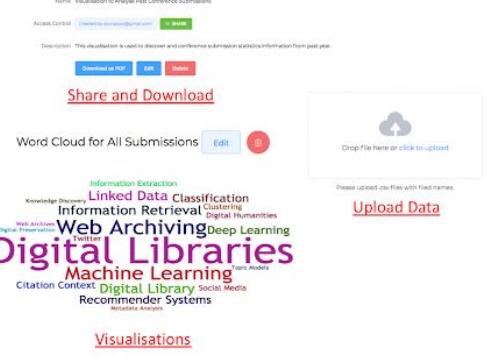
A step by step general guide to use the system:

FeatureGuide.vue

1. Log in using your Google account through the "Login" button.
2. Upload any number of csv files under the "Import Data" tab.
3. Create a new presentation and add visualizations using the data uploaded.
4. Share the presentation to your fellow friends to view or edit.



Google Log-in Page



Share and Download

Word Cloud for All Submissions

Upload Data

Visualisations

Currently, these are the pre-defined features that we support:

- Word cloud for all submissions keywords
- Submission rank author
- Submission rank country
- Submission rank organization
- Accepted submission organization rank
- Review Score Distribution
- Review weighted evaluation score statistic summary
- Reviewer Expertise Level Statistic Summary
- More options can be found under the presentation pre-defined dropdown.

More Detailed User Guide

DetailedFeatureGuide.vue

- **User Guide for Importing Data:**

- Step 1. Upon logging in, click on the "Import Data" tab.
- Step 2. Upload or drag the csv file from your computer.
- Step 3. Choose the table types of the csv file uploaded (i.e. Author Record, Review Record, Submission Record).
- Step 4. Select if there is a header row in the uploaded csv file.
- Step 5. For your selected table type, map the database fields to the imported data fields one by one.
- Step 6. Finally click upload to update the database accordingly.

◦ **Note:** You can select any number of columns to map into the database. Please map all the data fields you need.

- **User Guide for Adding Presentation:**

- Step 1. Upon logging in, click on the "Analyze" tab.
- Step 2. Create a new presentation name (minimum 3 characters) with description (optional).
- Step 3. Select a list of pre-defined visualizations for submission, author or review.
- Step 4. Click "Add New Section" to view the selected visualization section.
- Step 5. For each pre-defined visualization, you can edit, delete, filter and preview.
- Step 6. In Edit Mode, users can:
 - a. Change the name and add description of the visualization.
 - b. Edit the mode of visualization.
 - c. Add filter for the field and value to hide from the visualization.
 - d. Preview the changes before making the final change through the save button.
- Step 7. Delete - Remove the visualization section from the presentation.
- Step 8. There is no limit to the number of visualizations to added by the users.

◦ **Note:** The default mode for users is basic mode. Advanced mode is for users to make SQL-like queries straight from the database. We will update a separate documentation on this in the later stage.

- **User Guide for Sharing Presentation:**

- Step 1. Click on share button.
- Step 2. Enter the email address of the user to share to.
- Step 3. Change the permission (view or edit) given to the user.

◦ **Note:** There is also an option to give Public Access Control (view or edit). And you can view who has access to the presentation as well.

Figure 17: Breakdown of Home view

Home Analyze Import Data Logout (pengcheng.3a1) MenuBar.vue

Presentations created by me

ListOfPresentation.vue

- New
- Submissions Visualisation
- Authors Visualisation
- Reviews Visualization
- Mix Visualization author sub
- mix visualization review sub
- mix visualization author rev
- General Visualizations

Name: General Visualizations PresentationBrief.vue

Access Control: Created by pengcheng.3a1@gmail.com **SHARE**

Description

Download as PDF **Edit** **Delete**

SectionListPanel.vue

Please select a section to add **Add New Section**

AbstractSectionDetail.vue Word Cloud for All Submissions **Edit** **Print**

WordCloudSectionDetail.vue BasicSectionDetail.vue

Classification
Citation Context Linked Data
Information Retrieval Topic Models
Deep Learning Twitter Web Archiving
Digital Preservation Clustering
Digital Humanities Web Archives
Machine Learning Knowledge Discovery
Digital Library Information Extraction
Recommender Systems

This word cloud shows a list of key words found under the abstract section for all the submitted papers.

AbstractSectionDetail.vue Submission Rank Paper Author **Edit** **Print**

BarChartSectionDetail.vue → BasicSectionDetail.vue

Author	Paper Counts
V. Steinbock	6
Diana Baloozoot	5
Ariboor Makhsoos	4
Brix Grix	4
Conor Broderick	4
Jar Hazz	4
Microno Novis	4
Microno Wexx	4
Sherlock Deck	4
Ado Japoot	3

■ Paper Counts

This bar chart shows the number of papers submitted by each author in descending order. This tells us which authors submitted more papers than other authors. As this is not an absolute number, it is calculated by splitting the authors field in each submission into individual authors and aggregate the number of submissions for each author.

AbstractSectionDetail.vue Submission Rank Country **Edit** **Print**

PieChartSectionDetail.vue BasicSectionDetail.vue

Country	Percentage	Count
United States	32.05%	200
China	16.99%	106
Germany	15.54%	97
India	7.21%	45
United Kingdom	3.21%	20
Japan	2.72%	17
Ireland	2.24%	15
Canada	2.08%	13
Other	1.41%	9
Netherlands	1.26%	8
New Zealand	1.26%	8

This pie chart shows the percentage and number of papers submitted from each country. This tells us which country has more submissions than other countries. We have included others to account for all countries involved.

Figure 18: Breakdown of Analyse view

The screenshot shows a 'Mapping' section where database fields are mapped to imported data fields. The database fields listed are: Submission Id, First Name, Last Name, Email, Country, Organisation, Web Page, Person Id, and Is Corresponding. The imported data fields listed are: submission #, first name, last name, email, country, organization, Web page, person #, and corresponding?. Each mapping is represented by a row with a delete 'X' icon on the right.

Figure 19: Breakdown of *ImportData* view

Figure 20 shows the breakdown of a specific visualization. The component outlined in blue are controlled by **BasicSectionDetail.vue**. The components outlined in orange are **BarChartSectionDetail.vue**. All the logic related to the title, description and filtering of the section are handled in the **BasicSectionDetail**. **BarChartSectionDetail** only cares about how to generate visualization based on given data. Such separation of concern tremendously increases the reusability of the frontend components. Any new type of visualization could be added with the help of **BasicSectionDetail**. Any existing visualization will remain untouched (Open–Closed Principle).

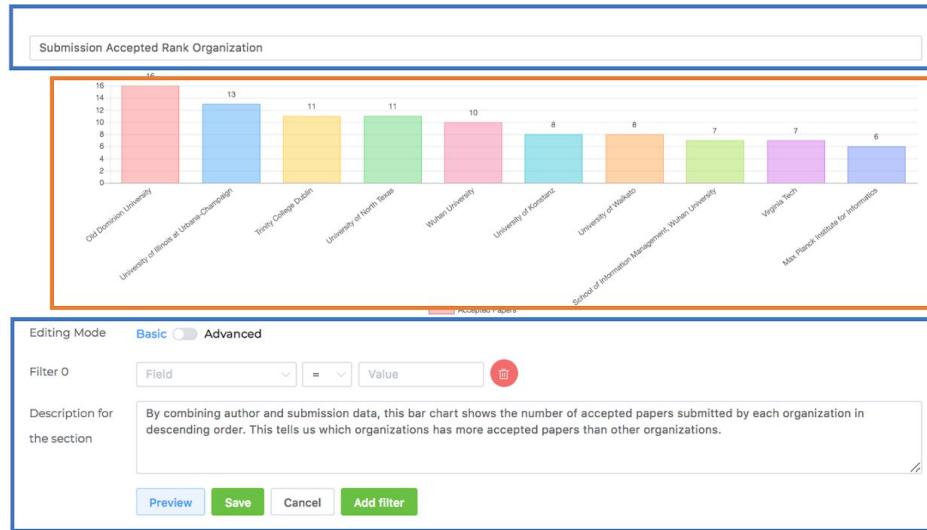


Figure 20

It should also be noted that the component **BarChartSectionDetail** is not specific to show “Submission Accepted Rank Organization”. Any visualization that is displayed in bar chart format can reuse the component.

Code Implementation

This is a step-by-step guide for setting up a development environment on your local machine. Using this environment, you can contribute to the project by working on features, enhancements, bug fixes, etc.

Getting Started

To get started, you should have installed the latest version of IntelliJ on your local machine. Then please follow the steps.

Get a Copy

Clone the repo to your local machine.

Environment Setup

- Config your IntelliJ. Please install the following plug-ins:
 - Google Cloud Tools
 - Vue.js (optional)
- Config your Google Cloud SDK. You can follow the instructions <https://cloud.google.com/sdk/>
 - Prepare for app engine deployment under project root directory:
`$ gcloud -q components install app-engine-java`
 - Config cloud SQL proxy. Please follow the instructions <https://cloud.google.com/sql/docs/mysql/connect-admin-proxy>. The application will use Google Cloud SQL as a database. It is also possible to set up an local MySQL running instance
 - Install npm. We suggest using NVM, which stands for Node Version Manager. You can follow instructions <https://www.npmjs.com/get-npm>
 - Install Sequel Pro from <https://www.sequelpro.com/> (optional)
- Configuration setup (optional). By default there are pre-set configurations for the project. However, if you may wish to change them, you can edit the following files:
 - `src/main/resources/application-local.properties` for SQL server connection setup (If you are using your own MySQL server, you need enter the correct connection string here)
- Run application backend
 - Import the project into IntelliJ
 - Inside IntelliJ you can select run configuration as Google App Engine Standard Local Server
 - Click run button (a green arrow) to run the application
 - It is also possible to run the application by issuing cmd `$./gradlew appengineRun` without importing the project
 - Access the application through <http://localhost:8080>
- Run application frontend
 - Navigate to `src/web/app`.
 - Install the dependency use `npm run install`.
 - Run `npm run serve`
 - Access the frontend application through <http://localhost:4040>

Test and Deploy

Test

Here are two main components in this project's testing stage; front-end testing and backend testing.

- Frontend tests:
 - Make sure that npm is properly installed
 - Run the following commands under root directory.
 \$ cd src/web/app
 \$ npm run test:unit
- Backend tests:
 - Run the following commands under root directory.
 \$./gradlew check

Deploy

You have two ways of deploying: locally and through continuous deployment.

- Local deployment:
 - Make sure that npm is properly installed on your local machine. You can verify it by calling
 \$ npm -v
 - Make sure that Google Cloud SDK is properly installed on your local machine. You can verify it by calling
 \$ gcloud --version
 - Authorize your Google Cloud SDK locally through
 \$ gcloud auth login
 - Make sure application-prod.properties contains correct configurations
 - Modify the template with corresponding information.
 - Build the frontend for production
 \$ cd src/web/app\$ npm run build
 - Go to project root directory and run
 \$./gradlew appengineDeploy
- Continuous deployment: Every time when master branch is updated, the new stable version will be automatically deployed to Google Cloud.

Suggestions for Future Improvements and Enhancements

Project Based Dataset

Currently, the application stored dataset for each user rather than each presentation. A user can import data and create multiple presentation based on the same data set. It would be great for user to import data for each presentation. For example, a user may want a presentation showing data insights generated for 2017 and a presentation showing visualizations generated for 2018. Due to time limit, our group do not implement this.

More Detailed User Guide

A completed user guide has been provided for the basic mode when he/she editing the sections. For power users, they may want to explore the advance mode when editing section. In fact, advance mode gives them the full flexibility of generating data by themselves. They can even create new visualizations. However, due to time constraint, such detailed user guide is lacking.

Merging Multiple Visualizations into one Graph

Based on the data, the graph produced for each visualization is standalone. It will be tedious for users that want to combine different visualizations together to see a comparison. For example, users may want to combine average confidence and expertise level of reviews side by side to see the relationship between confidence and expertise. Hence, a future implementation to combine graphs together into one visualization will be helpful.

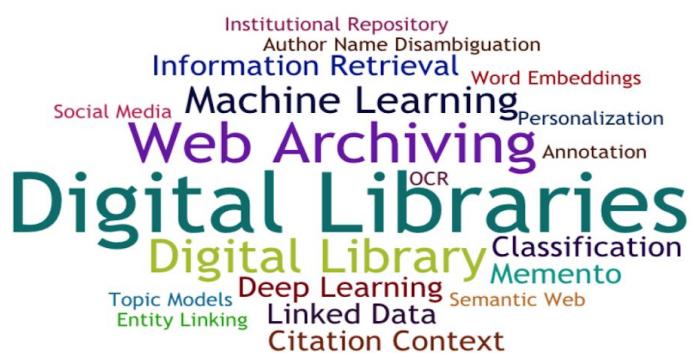
Adding Dropdown Options to the Filter Function

Users can input any details for filter function. However, the filter function only works if the user enters the correct label that correlates with the database. Not all user input may be able to be filtered. Hence, future implementation can considering adding a dropdown for users to select from a list of filters available to get a better visualization.

Visualization Screenshots

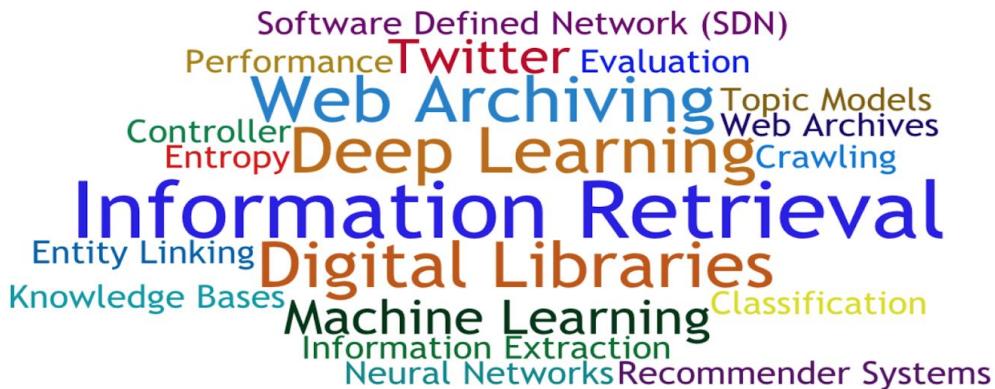
Submission Record Charts

Word Cloud for Accepted Submissions Keywords



This word cloud shows a list of keywords found under the abstract section for all the accepted papers.

Word Cloud for All Full Papers Submissions Keywords



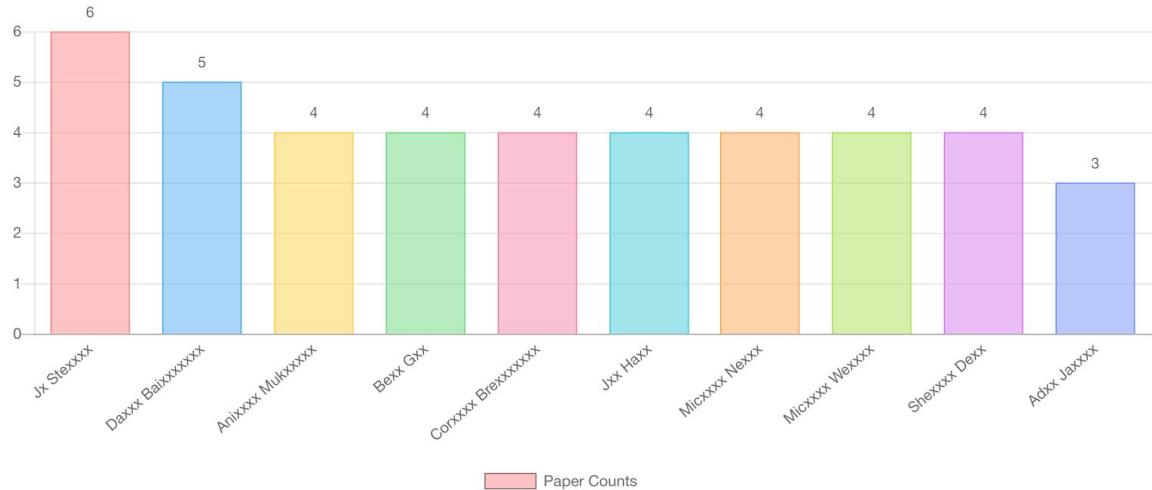
This word cloud shows a list of keywords found under the abstract section for all the submitted papers in Full Papers Track.

Word Cloud for All Submissions



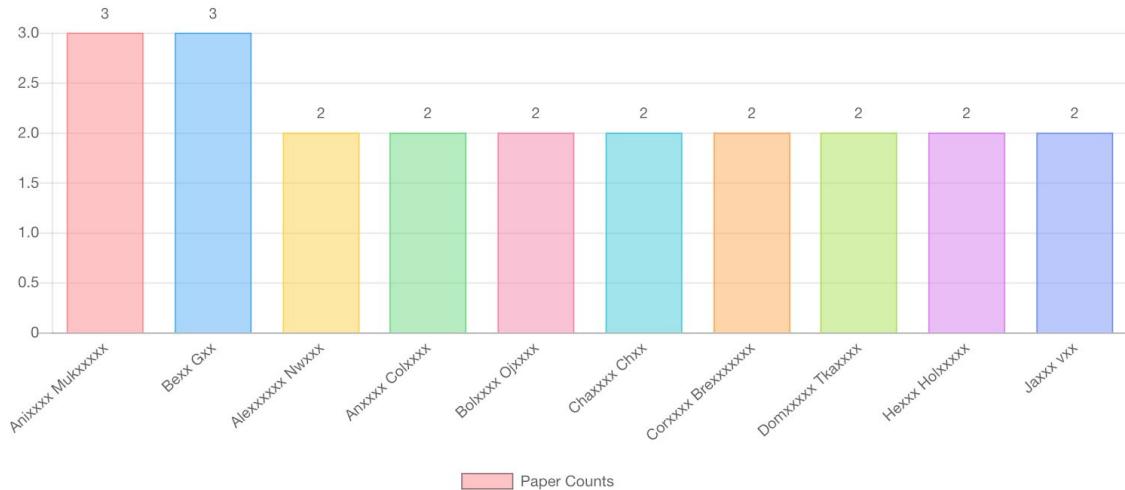
This word cloud shows a list of keywords found under abstract section for all submitted papers.

Submission Rank Paper Author



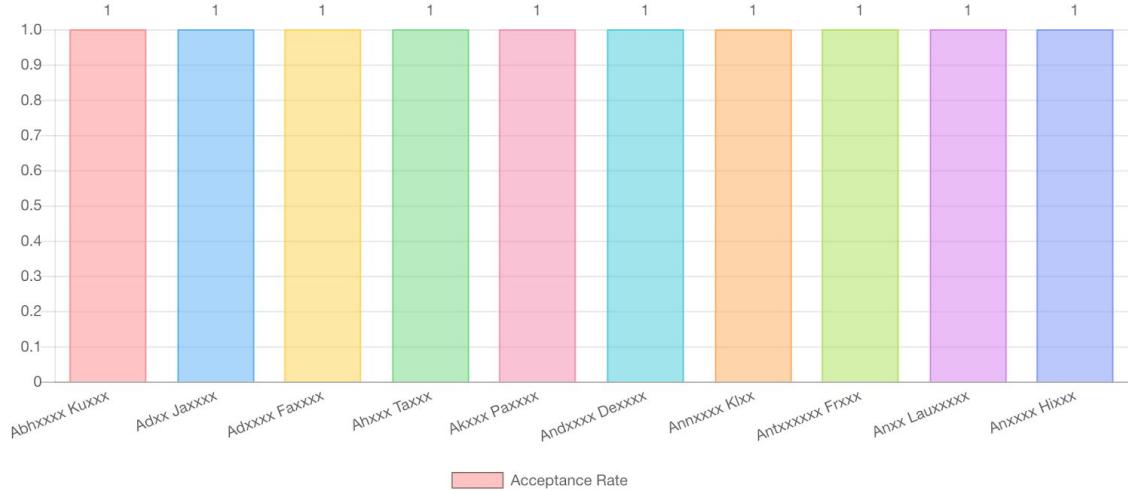
This bar chart shows the number of papers each author submitted based on descending order. This tells us which authors submitted more papers than other authors. We have split the authors field in each submission into individual authors and aggregated the submissions for each author.

Submission Rank Paper Author in Full Papers



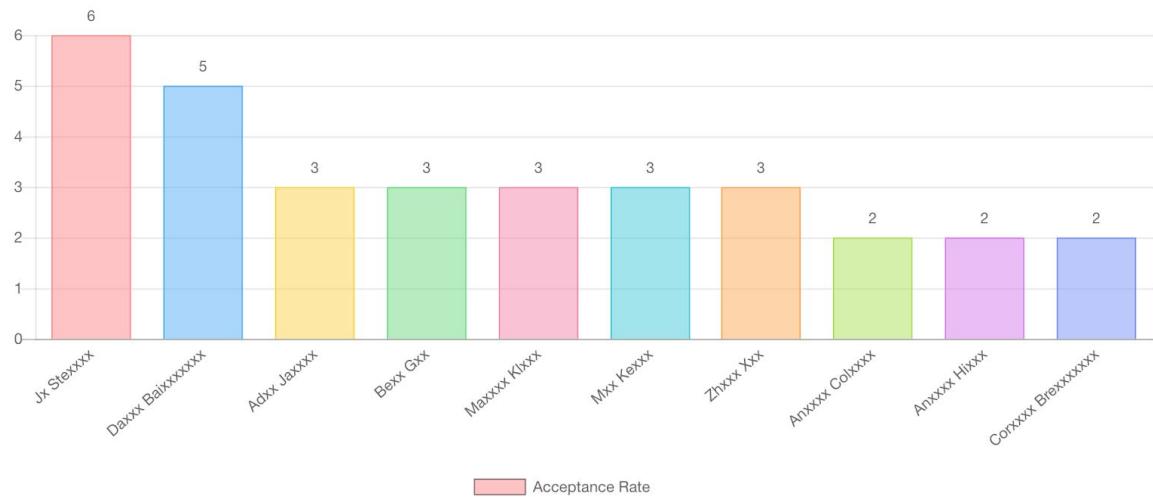
This bar chart shows the top accepted papers in the Full Papers track. This tells us the ranking of the authors in each individual track. Users can use the filter function to change to different tracks to view the top accepted authors in that track.

Submission Acceptance Rate Rank Paper Author



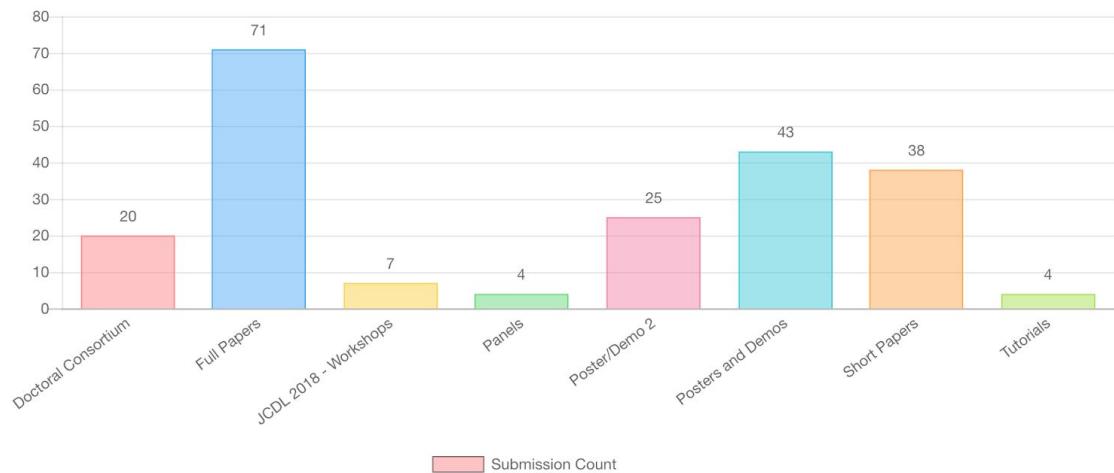
This bar chart shows the percentage of acceptance rate of each author's papers in descending order. This tells us which authors has higher acceptance rate than other authors. We have split the authors field in each submission into individual authors and calculate the acceptance rate for each author.

Submission Accepted Rank Paper Author



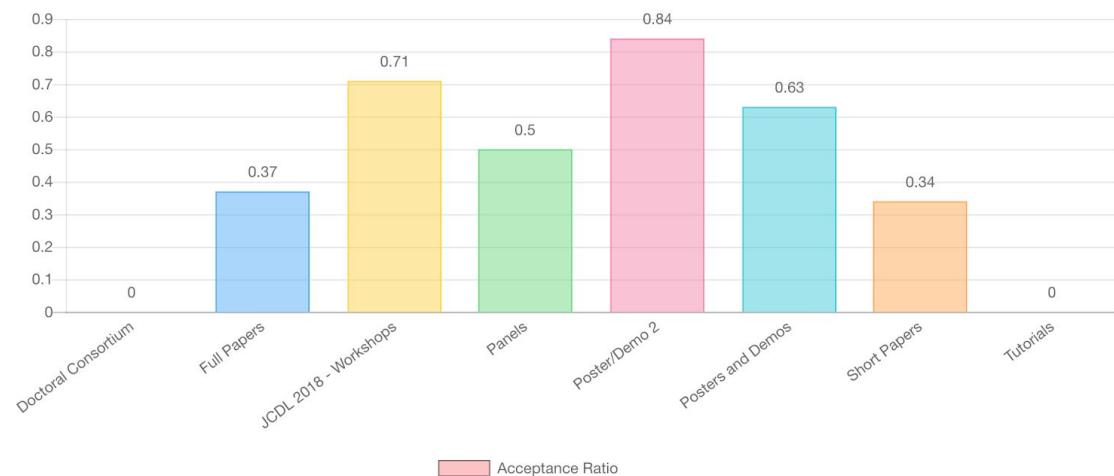
This bar chart shows the number of accepted papers submitted by each author in descending order. This tells us which authors has more accepted papers than other authors. We have split the authors field in each submission into individual authors and aggregate the accepted submissions for each author.

Submission Rank Track



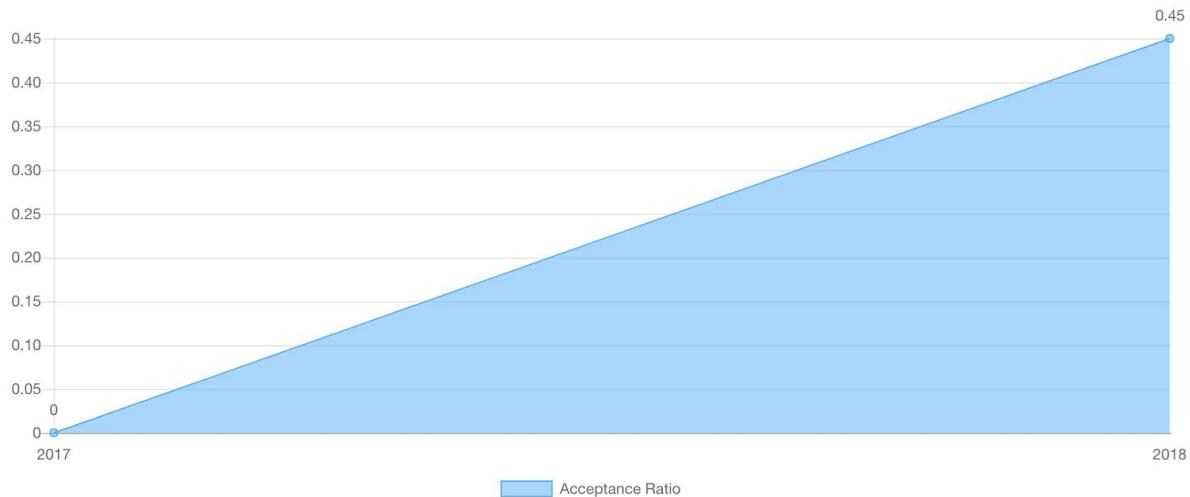
This bar chart shows the number of papers submitted in each track. This tells us which track is more popular than other tracks.

Acceptance Ratio Track



This bar chart shows the acceptance rate of each paper based on tracks. This tells us which track has higher acceptance rate than others.

Acceptance Ratio by Year



This area under line graph shows the acceptance rate of all submitted papers based on different years. This tells us which year has more papers getting accepted than other years. Note that data from years before 2018 was not provided and hence are all 0. To see trendings users might need to import data from those years.

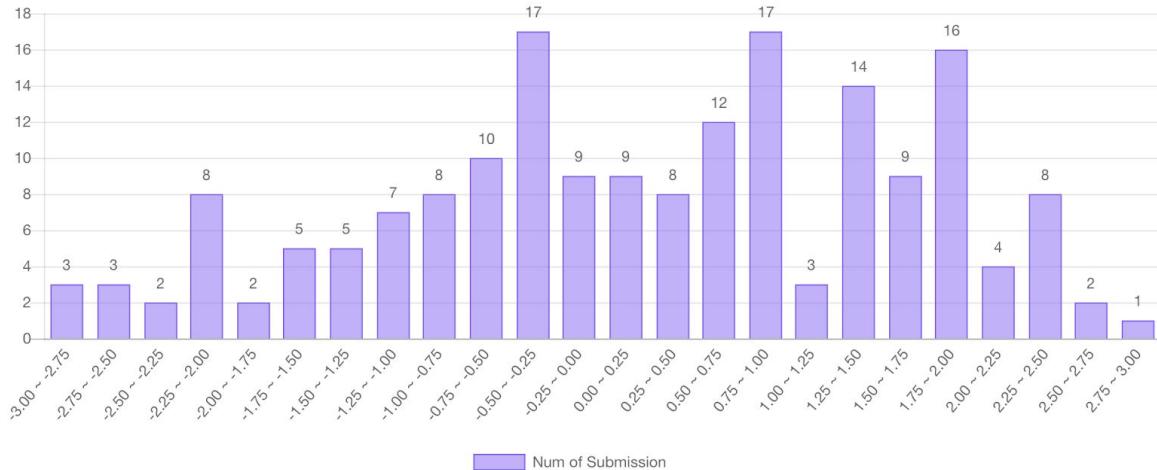
Review Record Charts

Word Cloud for Reviewer Comment



This word cloud shows a list of keywords found under reviewer's comments for reviewed papers.

Review Weighted Score Distribution



This bar chart shows the distribution of the weighted review score for all papers. This is calculated by retrieving the overall reviewed score and reviewer's confidence score in order to obtain a weighted average of all the scores. This gives us an insight on how the general score ranges for each submission.

Review Weighted Evaluation Score Statistic Summary

Type	Value
Min	-3
Max	3
Average	0.2706043956043956
Median	0.345
Standard Deviation	1.3863320914099144

This table shows the weighted evaluation score statistics based on the minimum, maximum value, the average, median score and the standard deviation of the weighted evaluation scores. This gives us an insight on the evaluation score given by each reviewer in their review.

Reviewer Expertise Level Statistic Summary

Type	Value
Min	1
Max	5
Average	2.168110918544194
Median	2
Standard Deviation	0.9962609829269724

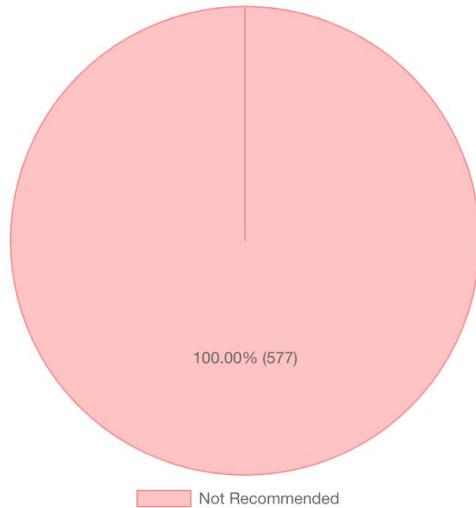
This table shows the reviewer expertise level statistics based on the minimum, maximum value, the average, median score and the standard deviation of the weighted evaluation scores. This gives us an insight on how specialized each reviewer is in their review.

Reviewer Confidence Level Statistic Summary

Type	Value
Min	1
Max	5
Average	3.653379549393414
Median	4
Standard Deviation	0.8058235933303965

This table shows the reviewer confidence level statistics based on the minimum, maximum value, the average, median score and the standard deviation of the weighted evaluation scores. This gives us an insight on how confident each the reviewer are in their review.

Recommendation for Best Paper Distribution



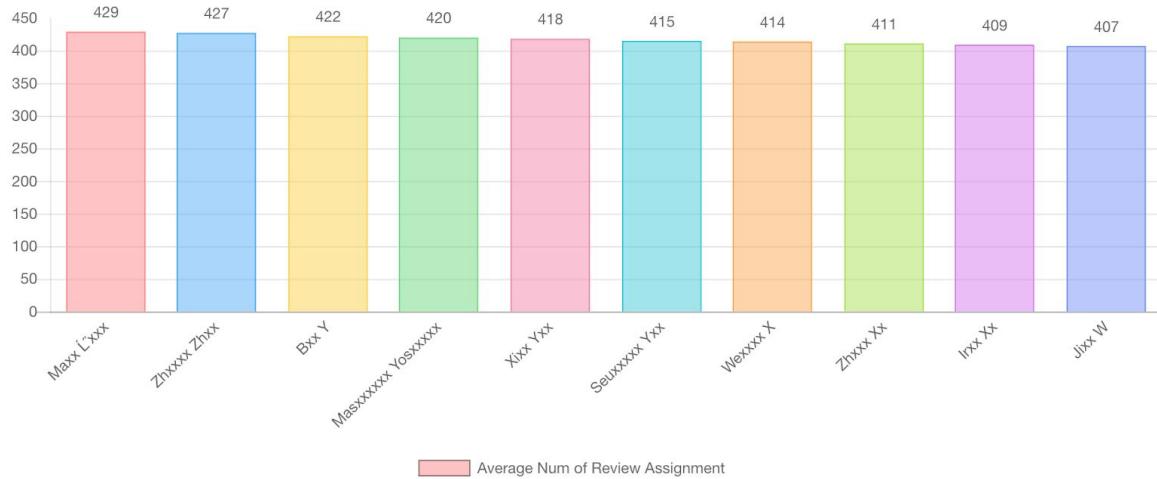
This pie chart shows the number and percentage of papers recommended and not recommended for best paper. This tells us how many papers are worthy to be submitted for the best paper. We have plotted into a pie chart as it is clearer to show a binary value.

Review Count Summary for Each Submission

Type	Value
Min	1
Max	5
Average	3.17032967032967
Median	3

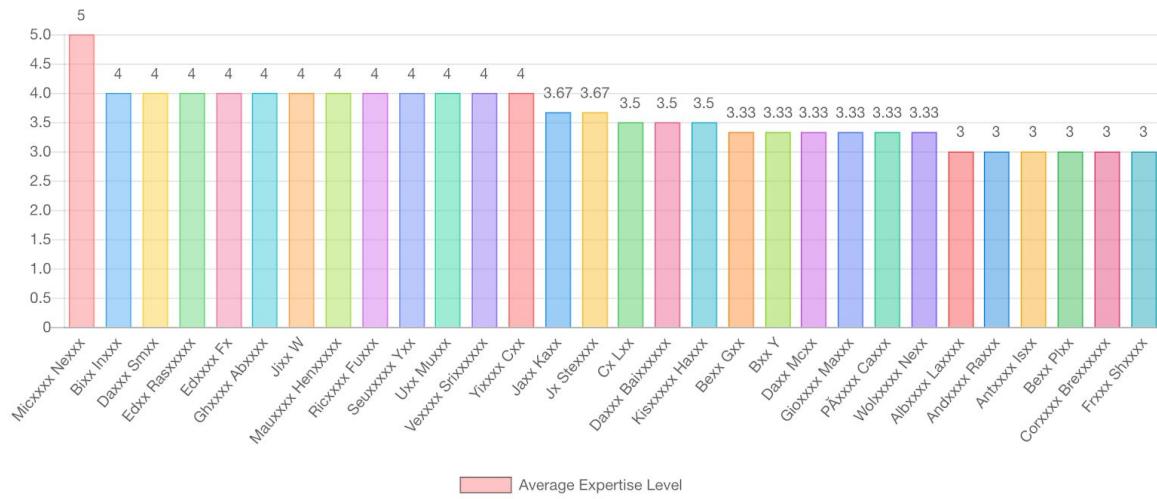
This table shows a statistic of the number of reviews for each submission based on the minimum, maximum value, the average and median score. This gives us an insight to how many reviews are provided for each submission.

Reviewer Assignment Rank



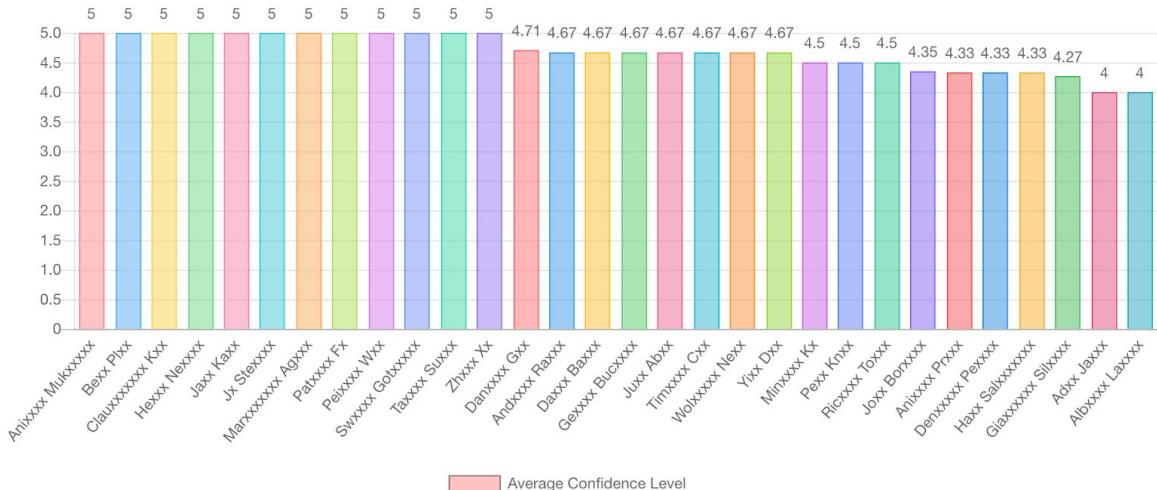
This bar chart shows the number of fields each reviewer is assigned to in descending order. A high number shows that the author is knowledgeable as he or she reviewed in multiple fields.

Reviewer Average Expertise Level Rank



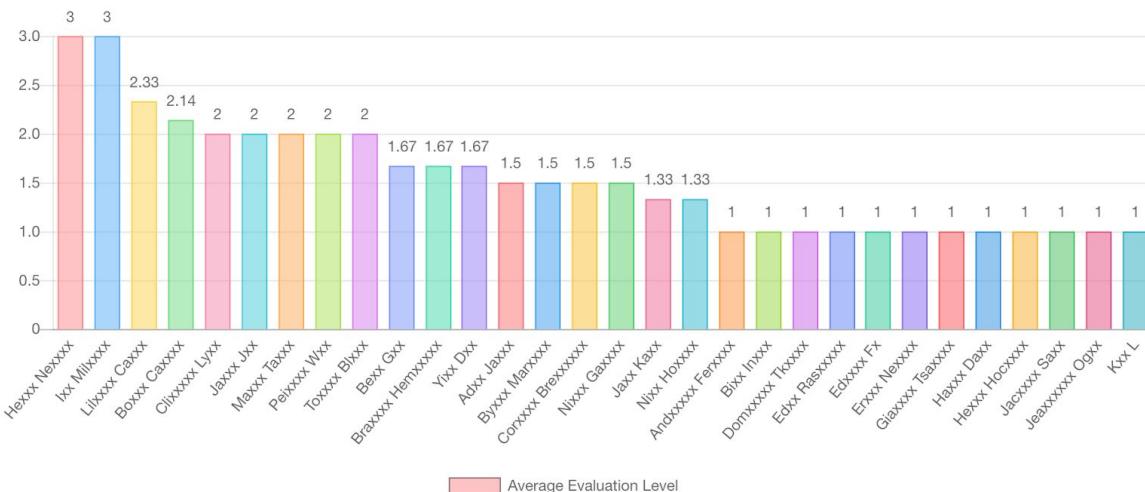
This bar chart shows the average expertise level for each reviewer in descending order. This tells us how skilled each reviewer is in reviewing the papers.

Reviewer Average Confidence Level Rank



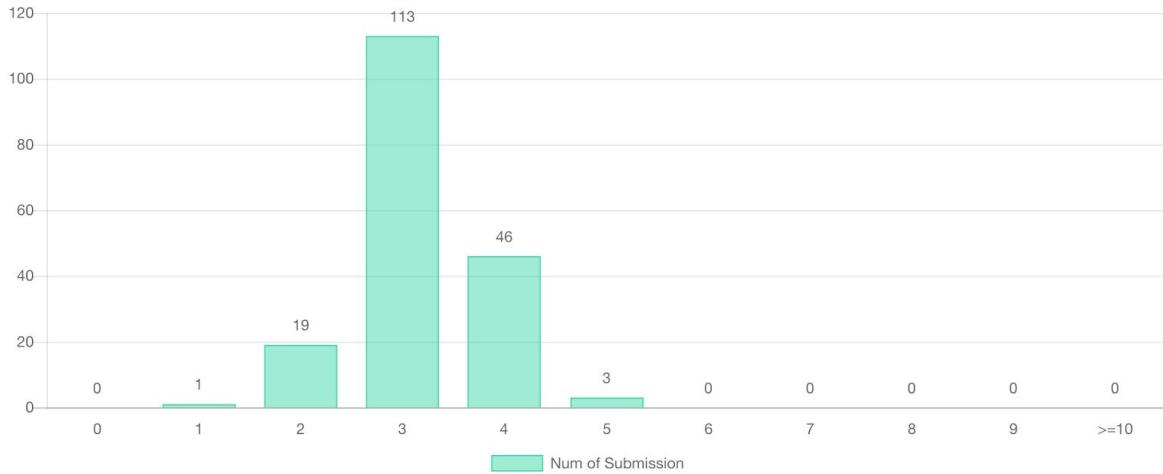
This bar chart shows the average confidence level for each reviewer in descending order. This tells us how confident each reviewer is in giving their review for other papers.

Reviewer Average Evaluation Score Rank



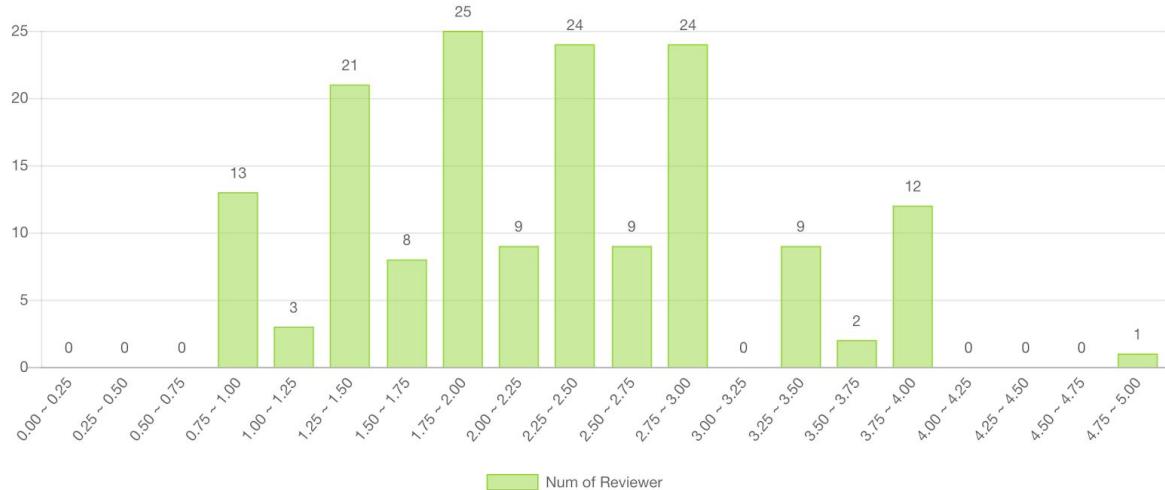
This bar chart shows the average evaluation score given by each reviewer in descending order. This gives us an insight of how generous the reviewer grade other papers in general.

Number of Review Distribution



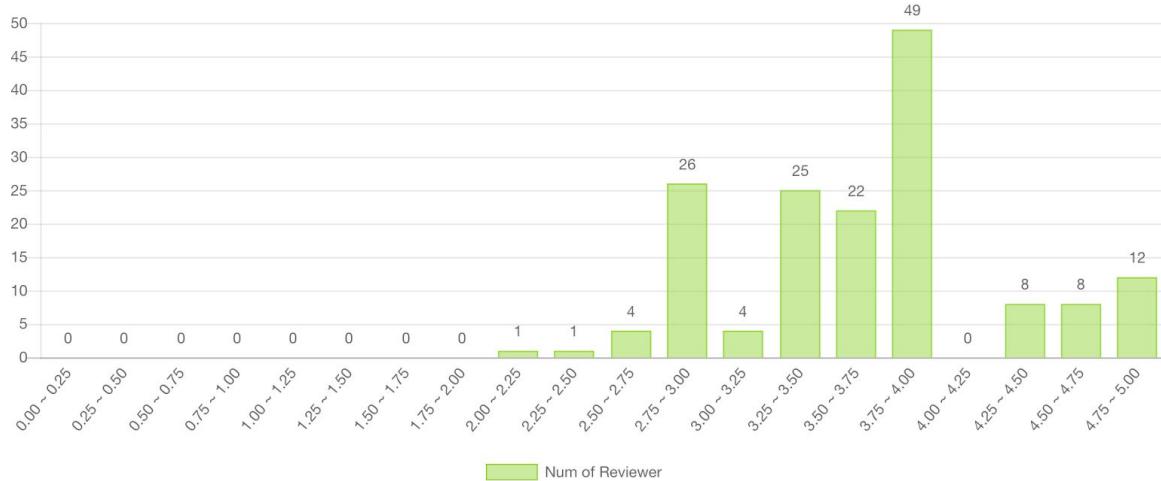
This bar chart shows the distribution of the number of reviews for each submitted paper. This gives us an insight on how many reviews are made for each submission.

Reviewer Average Expertise Level Rank



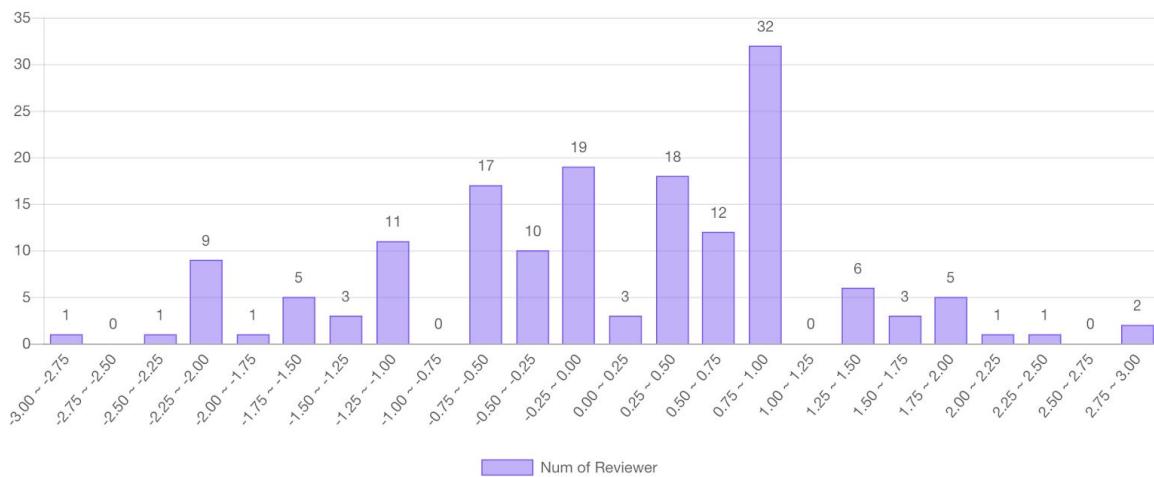
This bar chart shows the distribution of the expertise level of each reviewer in reviewing all the papers. This gives us an insight of how specialized each reviewer is in giving the review.

Reviewer Average Confidence Level Rank



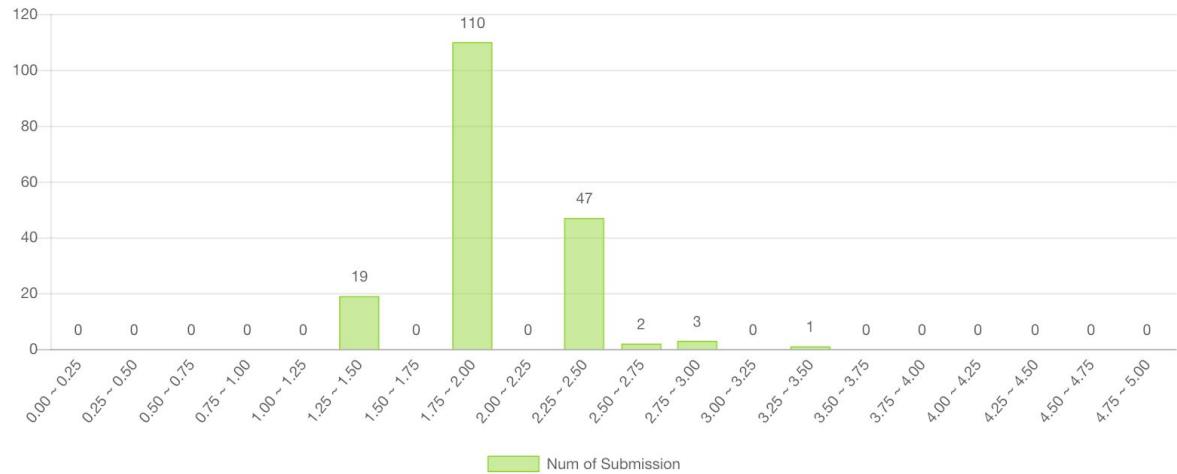
This bar chart shows the distribution of the confidence level of each reviewer in reviewing all the papers. This gives us an insight of how confident each reviewer is in giving the review.

Reviewer Average Evaluation Score Distribution



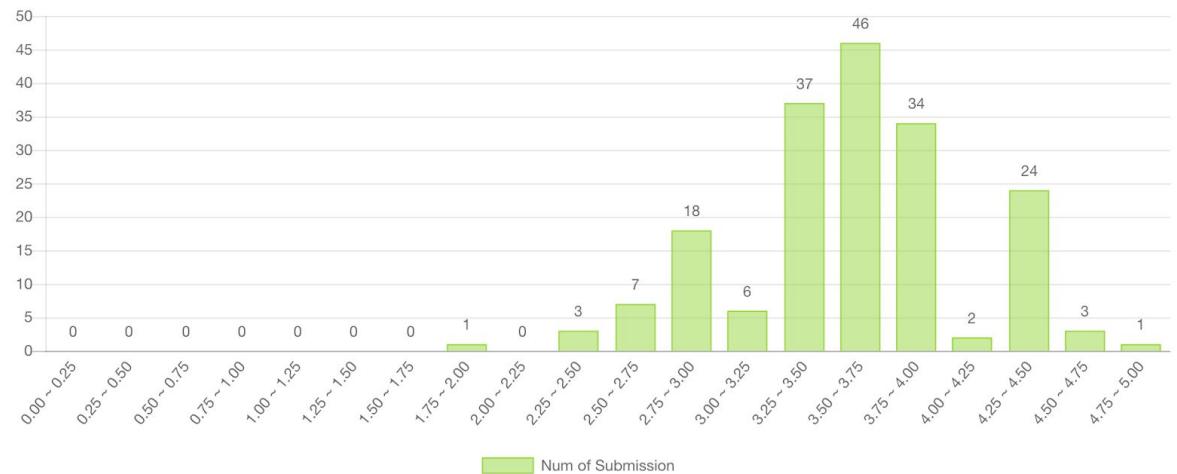
This bar chart shows the average evaluation score given by each reviewer in descending order. This gives us an insight of how generous the reviewer grade other papers in general.

Average Expert Level For Submission



This bar chart shows the number of submissions for different average expertise level. This gives us an insight on the average expertise level for different number of submissions.

Average Confidence Level For Submission



This bar chart shows the number of submissions for different average confidence level. This gives us an insight on the average confidence level for different number of submissions.

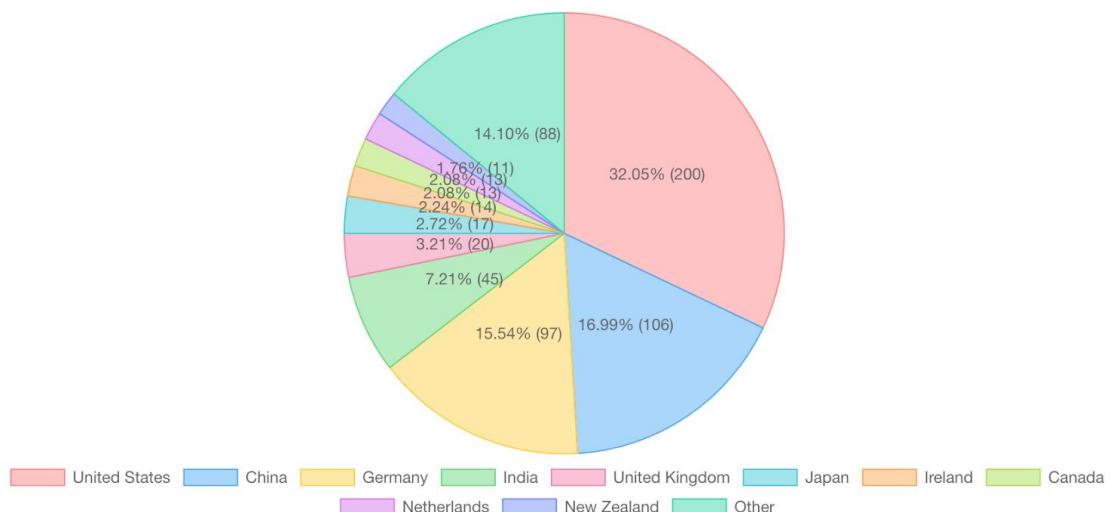
Author Record Charts

Submission Rank Author



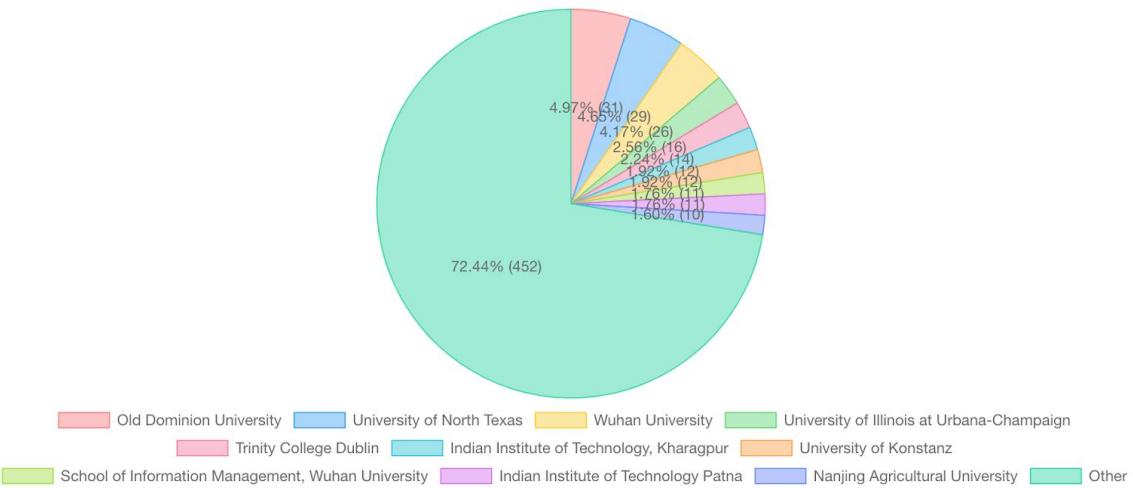
This bar chart shows the number of papers submitted by each author in descending order. This tells us which author has more submissions than other authors. Note that the uniqueness is determined by a pair of name and email address.

Submission Rank Country



This pie chart shows the percentage and number of papers submitted from each country. This tells us which country has more submissions than other countries. We have included others to account for all countries involved.

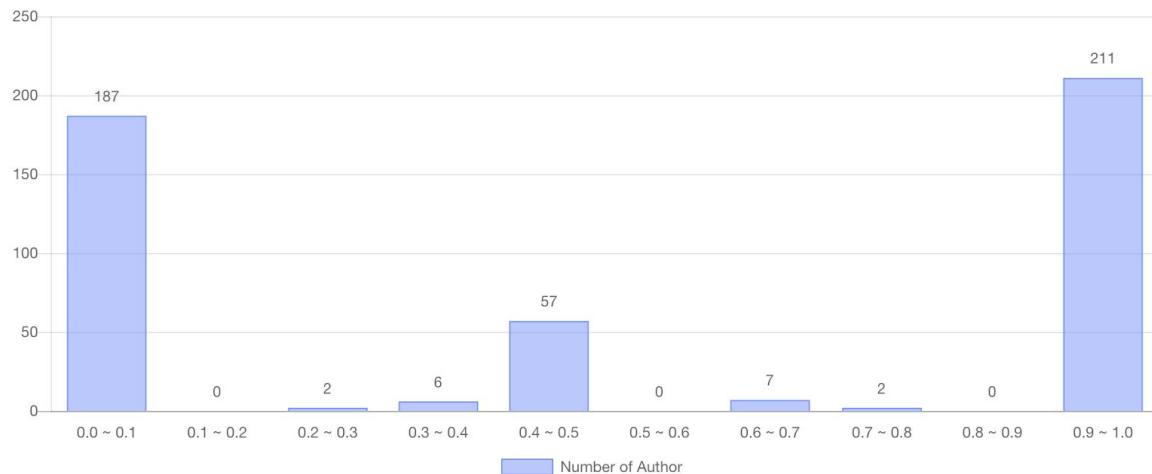
Submission Rank Organization



This pie chart shows the percentage and number of papers submitted from each organization. This tells us which organization has more submissions than other organizations. We have included others to account for all organizations involved.

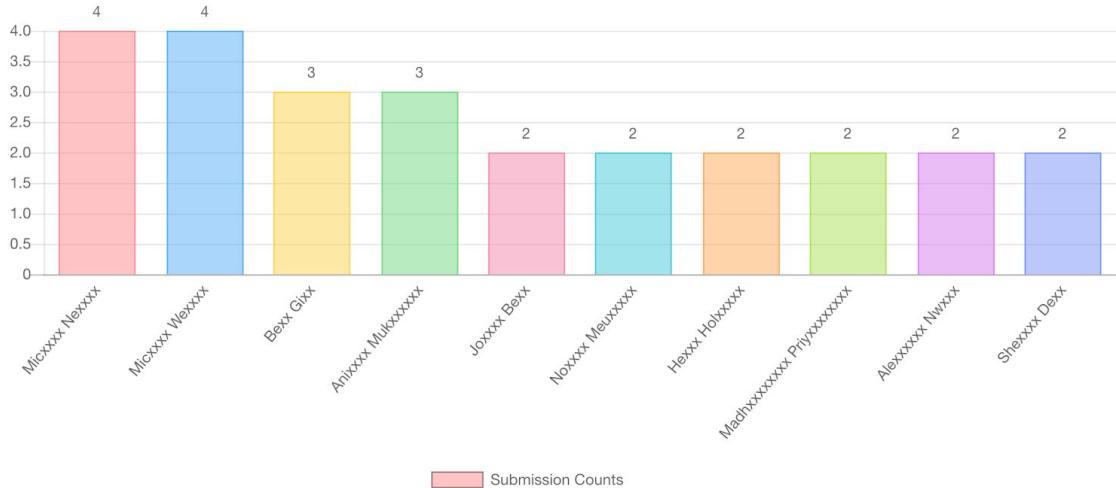
Submission + Author Record Charts

Submission Acceptance Rate Author Distribution



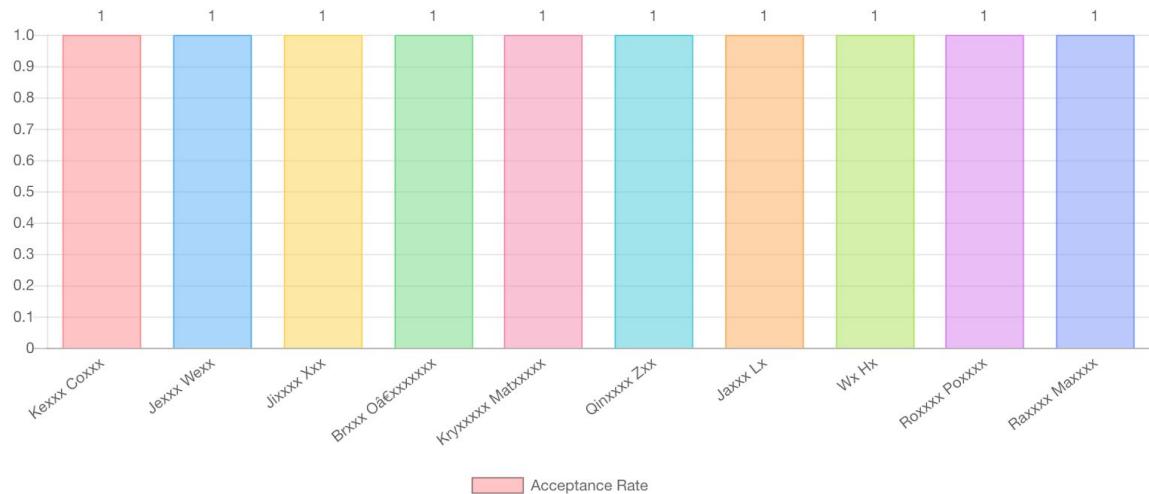
By combining author and submission data, this bar chart shows the distribution of acceptance rate for all authors. This tells us the capability of researchers who choose to submit in the conference.

Submission Rank Author in Full Papers



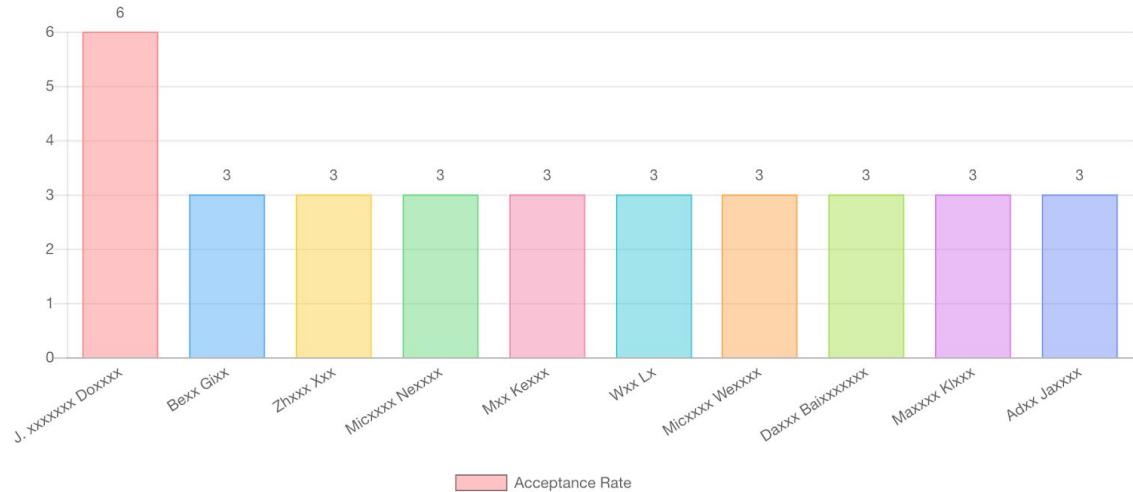
This bar chart shows the top accepted papers in the Full Papers track. This tells us the ranking of the authors in each individual track. Users can use the filter function to change to different tracks to view the top accepted authors in that track.

Submission Acceptance Rate Rank Author



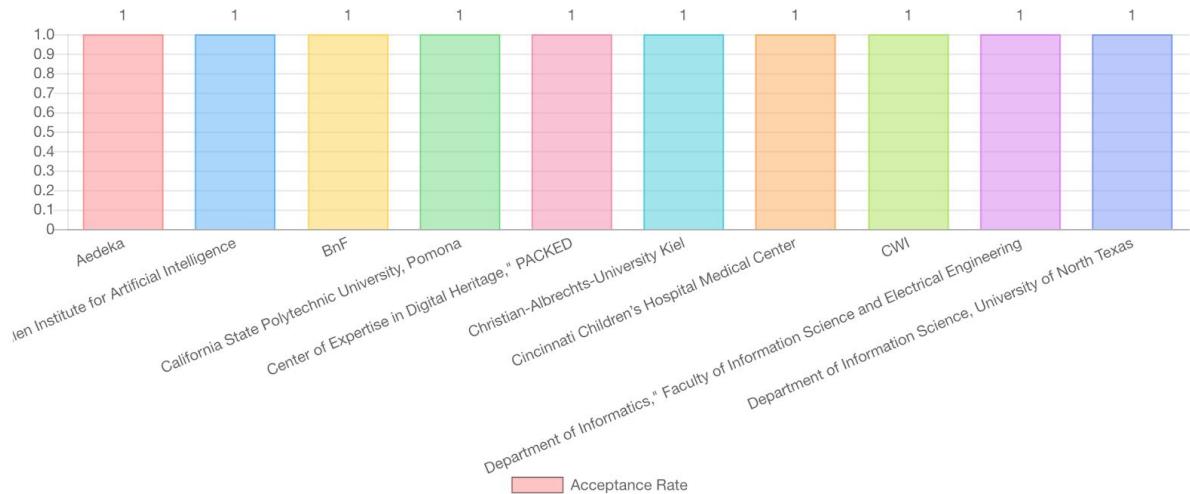
This bar chart shows the percentage of acceptance rate of each author's papers in descending order. This tells us which authors has higher acceptance rate than other authors. We have split the authors field in each submission into individual authors and calculate the acceptance rate for each author.

Submission Accepted Rank Author



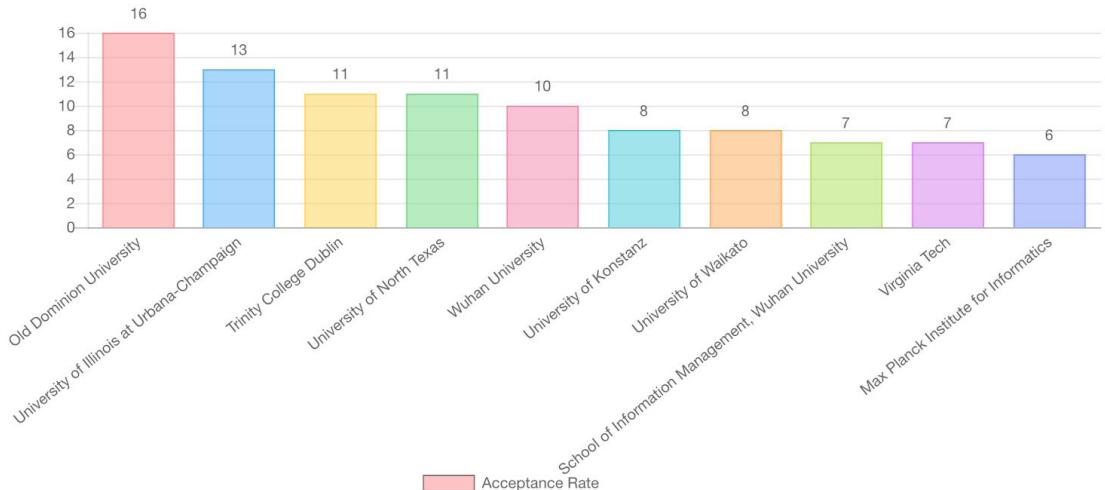
By combining author and submission data, this bar chart shows the number of accepted papers submitted by each author in descending order. This tells us which authors has more accepted papers than other authors.

Submission Acceptance Rate Rank Organization



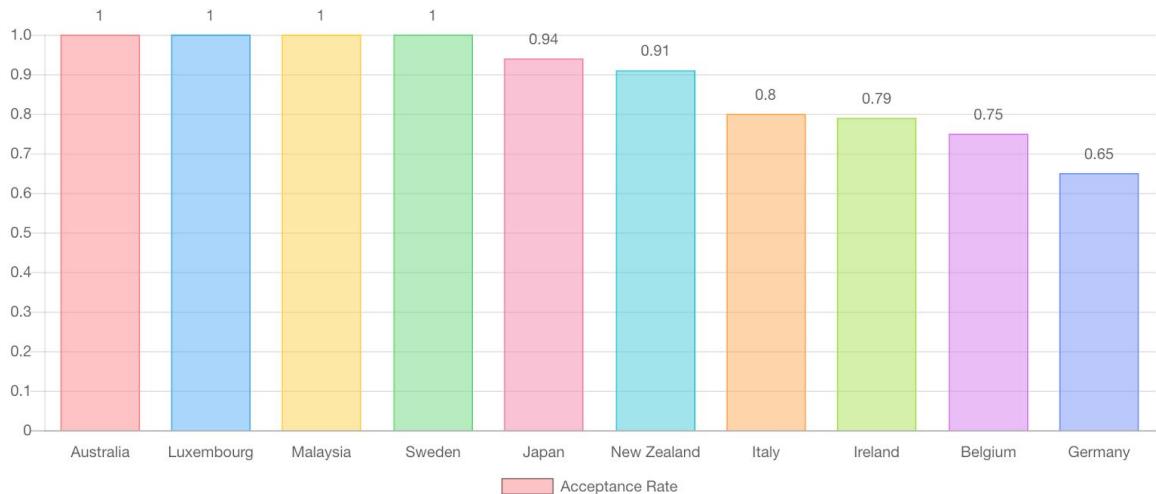
By combining author and submission, this bar chart shows the percentage of acceptance rate of each organization's papers in descending order. This tells us which organizations has higher acceptance rate than other organizations.

Submission Accepted Rank Organization



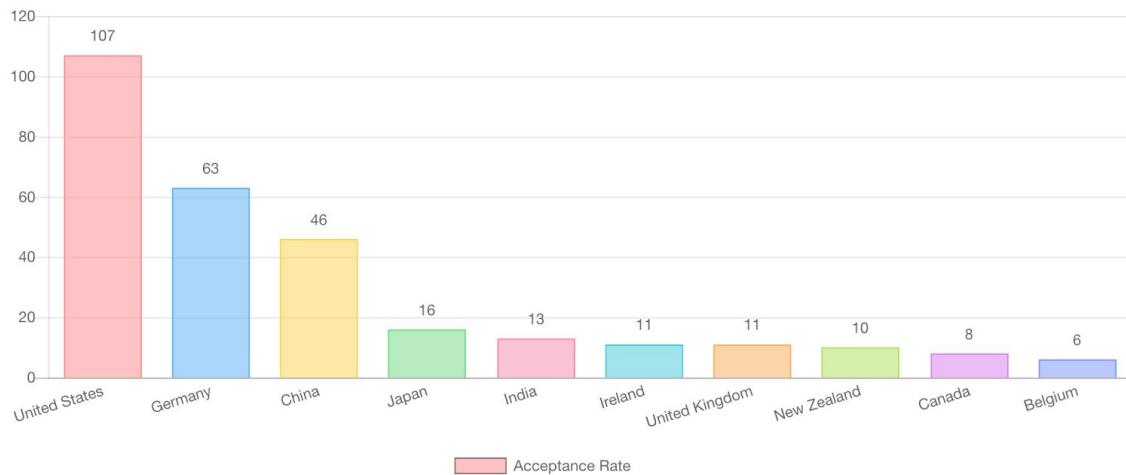
By combining author and submission data, this bar chart shows the number of accepted papers submitted by each organization in descending order. This tells us which organizations has more accepted papers than other organizations.

Submission Acceptance Rate Rank Country



By combining author and submission, this bar chart shows the percentage of acceptance rate of each country's papers in descending order. This tells us which countries has higher acceptance rate than other countries.

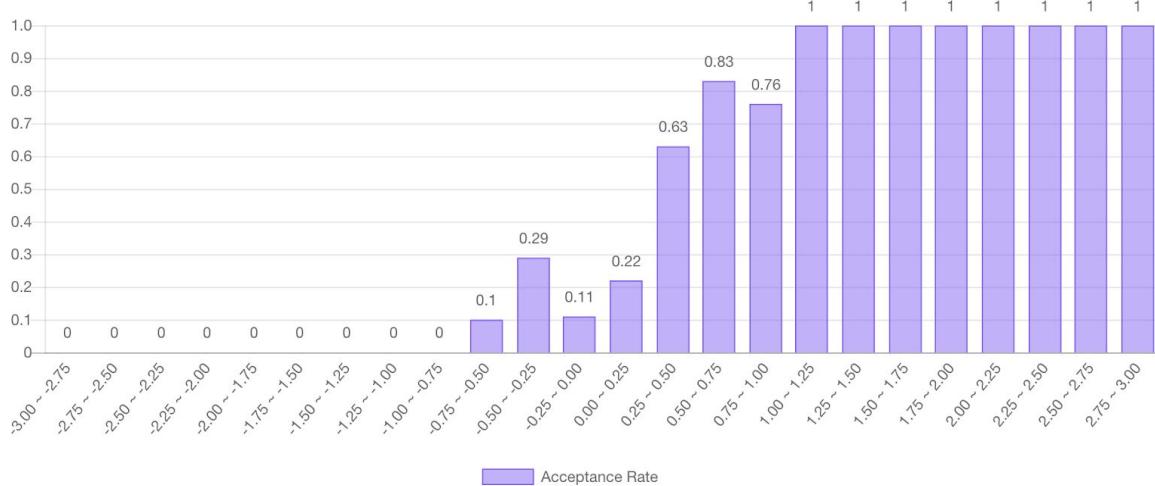
Submission Accepted Rank Country



By combining author and submission data, this bar chart shows the number of accepted papers submitted by each country in descending order. This tells us which countries have more accepted papers than other countries.

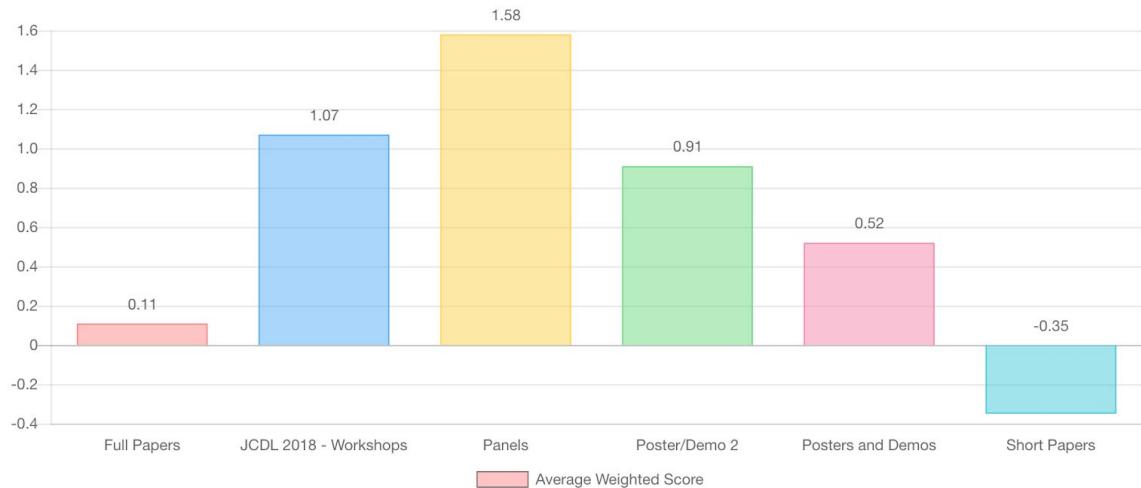
Submission + Review Record Charts

Acceptance Rate and Weighted Score



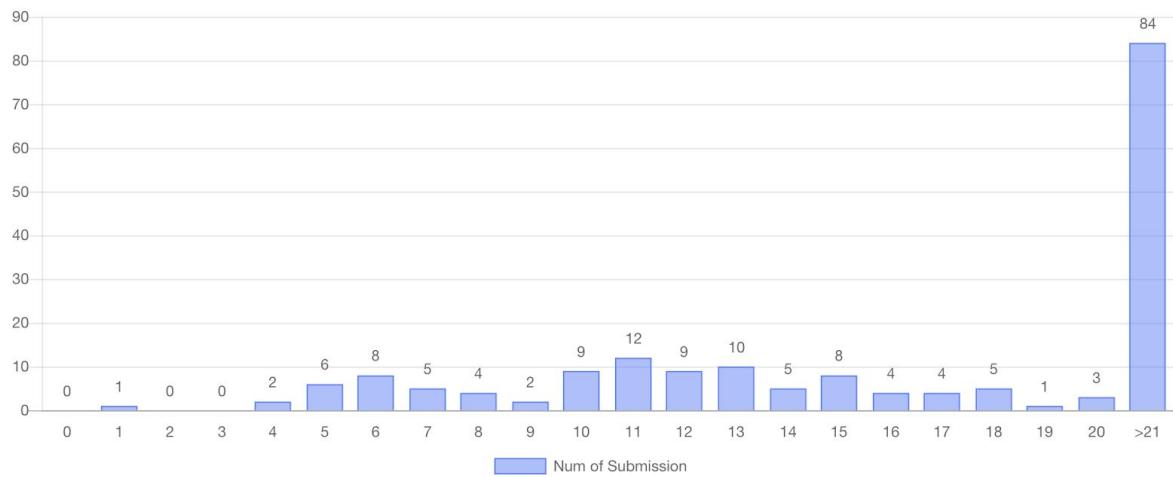
By combining review and submission, this bar chart shows the percentage of acceptance rate based on the weighted score of the submissions. This gives us insight on what percentage of the papers will be accepted given a review score.

Average Weighted Score By Track



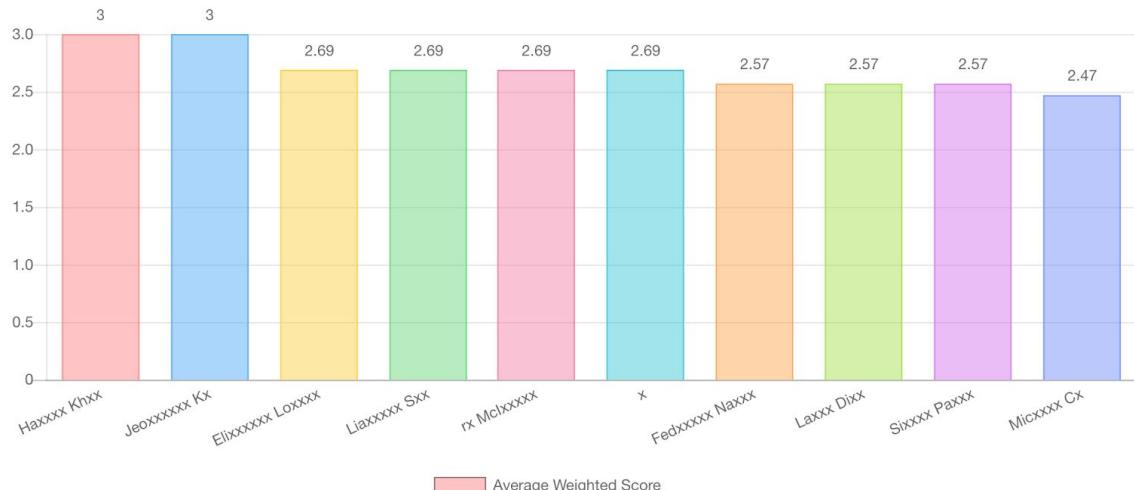
By combining review and submission, this bar chart shows the average weighted score for papers in each track. This gives us insight on which track has been weighted scores than other tracks.

Earliest Review in Days For Submission



By combining review and submission, this bar chart shows the number of days taken for submissions to be reviews. This gives us insight on the efficiency of the reviewers.

Average Weighted Score Rank Paper Author



By combining review and submission, this bar chart shows the average weighted score of all the submissions for each author. This gives us insight on which author performs better with his or her submissions than other authors.

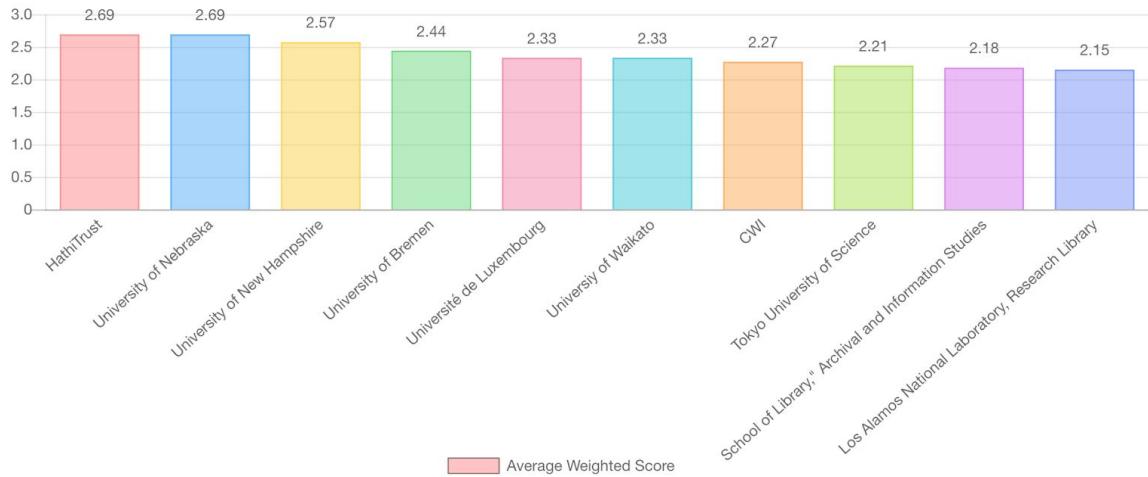
Author + Review Record Charts

Average Weighted Score Rank Author



By combining author and review, this bar chart shows the average weighted score of all the submissions for each author. This gives us insight on which author performs better with his or her submissions than other authors.

Average Weighted Score Rank Organization



By combining author and review, this bar chart shows the average weighted score of all the submissions for each organization. This gives us insight on which organization performs better with than other organizations.

Average Weighted Score Rank Country



By combining author and review, this bar chart shows the average weighted score of all the submissions for each country. This gives us insight on which country performs better with than other countries.