Department of Electrical, Computer, and Software Engineering

Part IV Research Project

Final Report

Project Number: #69

Measuring volunteer impact & success metrics in the Voluntarily platform

Harrison Leach

Project Partner: Jordan Sim-Smith

Project Supervisor: Andrew Meads

Date: 24/09/2020

Declaration of Originality

This report is my own unaided work and was not copied from nor written in collaboration with any other person.

Name: Harrison Leach

ABSTRACT: Voluntarily is an open-source online volunteering platform designed to connect volunteers and activity providers with schools or others in need. The platform has a commercial need to be able to measure and report volunteering impact and success metrics. This reporting system is essential to stakeholders for internal reporting and external promotion purposes. Therefore, this system is critical in maintaining stakeholder support for the Voluntarily platform. This paper explains the work undergone by the researchers in creating this system. A literature review was performed, outlining themes of volunteering metrics, survey design, data mining and data visualisation and highlighting their relevance to Voluntarily. The resulting implementation includes a system which allows users to view insightful volunteering metrics from a dashboard page. A feedback collection system has also been implemented, which further enriches the reporting capabilities of the platform. Future work includes extending the system to include metrics relevant to different parties, such as activity providers and schools. Additionally, the feedback system could be extended to elicit more detailed feedback from opportunity attendants.

List of Figures

1. Introduction

Voluntarily is an open-source volunteering platform that was designed to address the problem of misused corporate volunteer time and the need for improved teaching of digital content in New Zealand classrooms. Voluntarily aims to achieve this by bringing together three parties: "opportunity providers" (typically schools), "volunteer providers" (typically tech companies which provide corporate volunteers), and "activity providers" (providers of template activities and other teaching resources). The platform began as an initiative by the Pam Fergusson Charitable Trust, with backing from several companies, developers and designers.

In order to keep companies invested in the Voluntarily platform, they often need reports to provide to funders or management. Many companies have a corporate social responsibility (CSR) regulation [1] and may use corporate volunteering through Voluntarily to achieve this. Thus management needs some method(s) to assess whether or not volunteering through the Voluntarily platform is helping them achieve their CSR adequately. This research project aims to address this need by identifying and implementing suitable methods to measure and report volunteering impact and success metrics through the Voluntarily platform. Ultimately, this project is essential in keeping organisations invested in the platform and is thus also critical for the continuity of the Voluntarily platform.

Section 2 provides more background about the Voluntarily platform. Section 3 reviews the current state of literature relevant to the project and highlights valuable information and themes. Section 4 then builds on the literature review by explaining the statement of research intent. In section 5, the project requirements are explained, followed by a design overview in section 6. In section 7 the resulting implementation is explained with use-case examples, followed by a testing outline in section 8. In section 9, the various methodologies used throughout the project are explained and justified. Section 10 includes a reflection on the research intent, technologies and Voluntarily's project environment. Finally, section 11 summarises the achieved project goals and the future work needed.

1.1. Division of work

The major work items completed in this project include prototyping, architecture design and system implementation. All of these tasks were completed in tandem in remote working conditions. In the prototyping and design phases, work was completed using online tools that allow simultaneous participation such as Figma and Google docs. In the implementation phase, collaborative development was achieved through pair-programming sessions facilitated through Zoom. Meetings with the project supervisor occurred weekly during the semester, and meetings with the Voluntarily team occurred approximately monthly. The commitment and clear communication between the project partners resulted in a highly collaborative project with equally distributed work.

2. Voluntarily

The ministry of education in 2017, stated that it would be strengthening the focus of digital technologies in the New Zealand curriculum [2]. This decision has placed teachers in a difficult position as they typically do not have the necessary technical expertise to teach such content. Additionally, Voluntarily partners have identified, that while many companies have started charitable initiatives to offer corporate volunteering hours to employees, many of these hours are going unused. Voluntarily aims to solve these problems by bringing together three parties: “opportunity providers” (typically schools), “volunteer providers” (typically tech companies which provide corporate volunteers), and “activity providers.”

Other alternative solutions exist, such as HelpTank and the Community Comms Collective; however, they do not recognise activity providers as an essential party in this process. Furthermore, these solutions do not provide dynamic, insightful and easily accessible volunteering metrics. Such metrics are essential for the end-user to assess the success and impact of volunteering events. For Voluntarily stakeholders specifically, volunteering metrics are necessary for internal reporting and external promotion purposes. It is for this reason that this project is essential for the progression of Voluntarily. Moreover, it will contribute to the domain of online volunteering software in general.

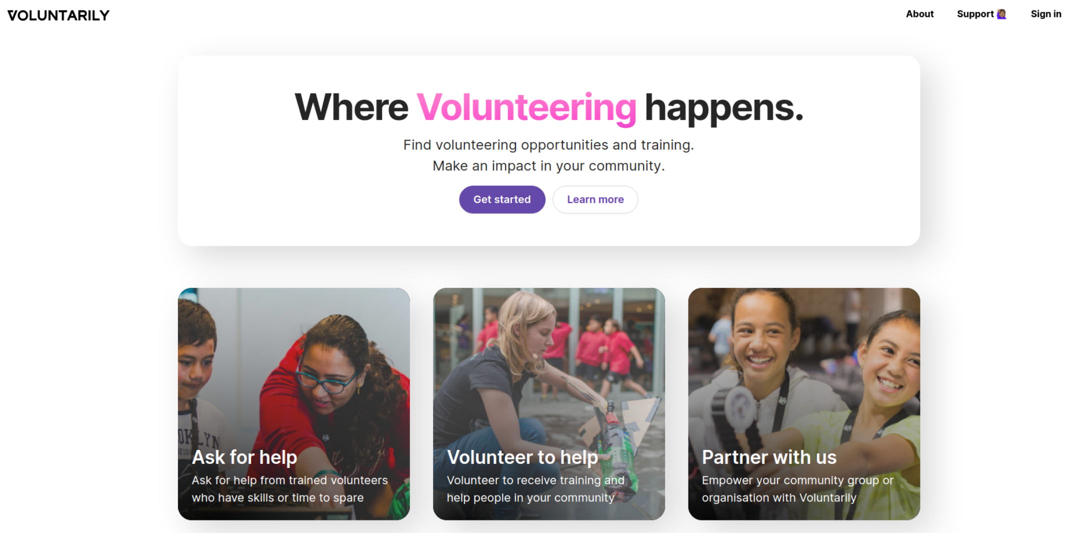

Figure 1: The Voluntarily platform home page

The Voluntarily platform (see figure 1) is built using the MERN stack of technologies. This stack consists of MongoDB[1] as the operational database, the Express JS framework[2] in building the back-end API, React.js[3] as the front-end framework, and Node.js[4] as the back-end runtime environment. Therefore, the Voluntarily platform is built using JavaScript exclusively. Since JavaScript is single-threaded, this implies some limitations in terms of data mining capabilities. Additionally, the Voluntarily platform currently has very little data and is unlikely to have a massive amount of data in the near future. Therefore, research regarding large scale data mining and analytics has been limited.

3. Literature Review

In determining the best solution for Voluntarily, through discussion with the product lead, the project's research domain was decomposed into four main areas: *i)* What metrics are useful to measure for volunteering impact and success? *ii)* What are effective methods for collecting such data? *iii)* How can the data be analysed and stored? *iv)* How can the data be visualised to present useful insights to the end-user? While there does not exist literature which answers these questions specifically in the context of a volunteering platform, the research has revealed metrics which drive corporate volunteering. Additionally, the research identifies techniques in effectively designing surveys to gather relevant data. In addition to relevant data mining methods and data

[1] `https://www.mongodb.com`
[2] `https://expressjs.com`
[3] `https://reactjs.org`
[4] `https://nodejs.org`

visualisation techniques and frameworks.

3.1. Metrics to measure volunteer motivation and engagement

The literature has repeatedly shown job characteristics are an important factor in driving volunteer participation [1] [3]. In a paper analysing the design of volunteer tasks [3], five job characteristics correlated with volunteer motivation were identified as useful predictors of volunteer motivation. These include:

- Skill variety - breadth of skills used in completing the task
- Task identity - how strongly the task is working towards a distinct goal
- Task significance - the impact of the task on other's lives
- Autonomy - freedom and independence in completing the task
- Feedback - performance review of the task.

From these, an overall Motivational Potential Score [4] can be derived:

$$MPS = (variety + identity + significance)/3 \times autonomy \times feedback \tag{1}$$

Jobs with high MPS are associated with positive outcomes such as internal motivation and higher levels of intrinsic satisfaction and job involvement [4]. The literature has also shown that volunteer involvement is influenced by the corporate volunteering program through which they are participating. Volunteer involvement is higher when volunteers perceive programs to be well-planned, seeking feedback, and reflecting organisational culture [1].

Other factors which influence volunteers motivation include: *i)* time and distance costs (having opportunities at a nearby location or being able to volunteer from home is most effective), *ii)* social connections (volunteers are more likely to volunteer if friends or family members are involved) and *iii)* information overload (volunteers can get overwhelmed by invitations to volunteering opportunities) [5]. In the context of Voluntarily, all of the factors and metrics mentioned above can be measured to evaluate the overall volunteer motivation and engagement levels for an opportunity, which can then be used to predict volunteer retention for future events.

3.2. Survey design

On the Voluntarily platform, there already exists sources for implicit metric collection (e.g. number of opportunities attended by a volunteer). However, as explained by of Health et al. [6] effective evaluation of volunteer programs (relevant for "activity providers") requires qualitative as well as quantitative data. Therefore forms of explicit metric collection such as surveys should also be used to measure volunteer impact and success.

In retrieving survey data from Voluntarily users, the results should be representative of all demographics on the platform. Thus, surveys should have a respondent-friendly design such that each person who is sent the survey has an equal chance of responding to it regardless of their computer self-efficacy or other factors. Additionally, respondent-friendly design should overcome four potential sources of error [7]: *i)* Coverage error *ii)* Sampling error *iii)* Measurement error and *iv)* Non-response error. Dillman et al. [8] described various principles for designing respondent-friendly web questionnaires, several of which are relevant to the Voluntarily platform. Firstly, questionnaires should begin with a brief motivational message explaining the purpose of the survey, as a long explanation or list of instructions is likely to result in a non-response. Additionally, the first question should be interesting, easily comprehensible and answered by all respondents. The first question should provide a positive first impression, so demographic questions should be avoided. In general, the questionnaire should give the user freedom by allowing them to skip questions.

Behr et al. [9] explained that web probes could be used to assess the validity of survey questions. Relevant probe types include category-selection, which ask respondents for their reasons in selecting a specific answer. This evaluates whether or not the respondent understood the categories correctly. Comprehension probing involves asking the respondent what they associate a certain term, which evaluates whether or not a respondent correctly understands a term as intended. Specific probing is useful for understanding how certain terms can be associated with different concepts.

Web probing can then be used throughout several stages of the release of a survey to assess its validity [9]. In the pretesting stage of a survey, Meitinger and Behr [10] suggest the use of cognitive interviewing, which could logically be followed by web probing. In the main production stage, it can be useful to have a split of respondents answering probes instead of all respondents to control for possible effects. Finally, in post-hoc evaluation, web probing can be used to shed light on existing data to explain anomalies in the data or to assess problematic questions in the data. Ultimately, once a respondent-friendly survey has been designed for Voluntarily, web probing can be used as an effective evaluation technique.

3.3. Data mining and data warehousing

Data mining, also known as knowledge discovery in databases, is the extraction of implicit, previously unknown and potentially useful information from databases [11]. Common data mining techniques found in the literature can be categorised into the following analysis types: association rules, data summarisation, classification, and data clustering. Association rules provide inferred statements about an entity, e.g. "If an entity does X, they usually do Y". Data summarisation provides a high-level overview of the data, which helps understand the structure of the data [11]. Data classification requires a classification model (set of classes) and involves assigning the correct classes to unlabelled objects [12]. Literature has shown churn prediction for online platforms can be modelled as a binary classification, (churners or not churners) using user lifecycle behaviours for classification of users [13]. Data clustering identifies densely populated regions based on some distance measurement, which uses the object's properties as variables [12]. Unlike classification, in data clustering, classes are not prespecified and are generated as the algorithm progresses [11].

The literature has shown that data warehousing is a desirable preprocessing step in performing data mining [12] [14] . In constructing a data warehouse, data is first extracted from operational databases; the data is then cleaned (e.g. correcting anomalies) and transformed if necessary; finally, the data is loaded into the warehouse [15]. Additionally, in using a data warehouse, there are typically three architectural components [12]: *i)* data acquisition software, *ii)* the data warehouse itself and *iii)* the client software (which allows users and applications software to analyse the data). A data mining application can be characterised as client software to a data warehouse. Data mining typically involves the use of data warehousing because data warehouses support online analytical processing (OLAP). As explained by Han et al. [16], the capability of OLAP to provide multiple and dynamic views of data sets a solid foundation for successful data mining. Typically operational databases are finely tuned to support online transactional processing (OLTP) as opposed to OLAP. Therefore mining data directly from an operational database would result in unacceptable performance [15]. In the context of Voluntarily, this means data warehousing may be required to provide acceptable performance for data mining.

3.4. Data visualisation techniques and frameworks

The literature has revealed various data visualisation techniques and where each may be useful. As explained by Ajibade and Adediran [17], line charts are useful in displaying multiple variable trends; an example is time series. A parallel coordinates plot takes data elements with multiple dimensions and maps each of their dimensions to their appropriate line along the x-axis. Parallel coordinates plots are effective in plotting data with many dimensions. Bar charts are useful for comparison of items from different groups, but should not be used when there are a huge number of values. Pie charts are a quick and effective technique of conveying comparison when percentages need to be compared. Pie charts are best when only a few components are compared. Scatterplots are a 2D plot displaying the joint variation between variables. Scatterplots show how strongly variables are related and are useful for discovering relationships.

In implementing the data visualisation techniques discussed above, many relevant JavaScript libraries are available. D3.js is the most popular and extensive JS data visualisation library, which leverages modern HTML, SVG and CSS web standards [18]. ChartJS is also a highly popular library which provides a much smaller and simpler collection of HTML 5 charts. Highcharts JS is potentially the most popular JavaScript charting API in the world, used by 72 out of the world's 100 largest companies. Metric-Graphics is a JavaScript library optimised for visualising time-series data; it also has a relatively small package size. Voluntarily is built with React, so a particularly relevant set of libraries is React Virtualized, React Vis and Victory, which exposes various charts and graphs as React components [18].

3.5. Literature Summary

Within the examined literature, some evidence is more applicable to the domain of Voluntarily than others. In section 3.1, the metrics provided by Richard and Oldham [4] are preferable to those provided by Kane and Klasnja [5] as Sekar and Dyaram [1] provide empirical evidence supporting the use of the former. In section 3.2, Dillman et al. [8] explains many useful survey design techniques which can be incorporated into the Voluntarily domain; however, some principles may not be achievable depending on Voluntarily survey design decisions (e.g. surveys may be done through third-party software). Furthermore, the use of web probing is very much dependent on the resources allocated for the implementation of surveys. In section 3.3, many useful data mining techniques are proposed. However, the use of data mining methods in Voluntarily may be limited due to the platform's use of JavaScript (which is single-threaded), and the lack of operational data. Data warehousing is still very relevant to the Voluntarily domain, as retrieving metric data will require complex queries which could benefit from OLAP. In section 3.4, all of the techniques and frameworks are relevant but ultimately depend on further Voluntarily requirements discussions.

4. Statement of Research Intent

This research project aims to identify and implement suitable methods to measure and report volunteering impact and success metrics through the Voluntarily platform. The solution implemented should provide a framework (a set of reusable components and examples) from which future developers can develop more metrics and graphs to extend the platforms reporting abilities. In the context of Voluntarily, reporting volunteering success and impact metrics provides organisations with feedback to improve their volunteering activities or practices. It can also be used to encourage future investment from management or funders. Ultimately, this project keeps the Voluntarily platform growing, helping it to achieve its goal.

In the academic context, throughout the discussion of the various concepts: Volunteering metrics, survey design, data mining and data warehousing and data visualisation, the literature had no mention of the use of the concepts within the domain of a volunteering platform. The research contribution will include an evaluation of these techniques within the volunteering platform domain and will discuss any advancements or adjustments needed (if any) due to the domain.

5. Requirements

5.1. Requirements Elicitation

Before designing a solution, additional user interviews were conducted with volunteer providers to help identify which features of the solution would be most valuable to them. From these interviews the following features of an MVP (minimum viable product) were identified: *i)* Show number of volunteers, total volunteer hours and average volunteer hours per employee across the company *ii)* Show what the volunteers volunteered for (by activity tag) *iii)* Show where the volunteers volunteered (by location). An extension to the MVP brought up later in the project was: *iv)* A feedback system that allows volunteers to provide a rating (1-5 stars) for an opportunity.

6. Design

As seen in figure 2, the design that has been created can be described in three parts. Firstly the front-end consists of the statistics page, the feedback page and the feedback email. Secondly, the newly created APIs consist of the statistics API and the feedback API. Finally, the database collections of interest include several existing collections and the newly introduced feedback collection. The design and implementation of the components is discussed further.

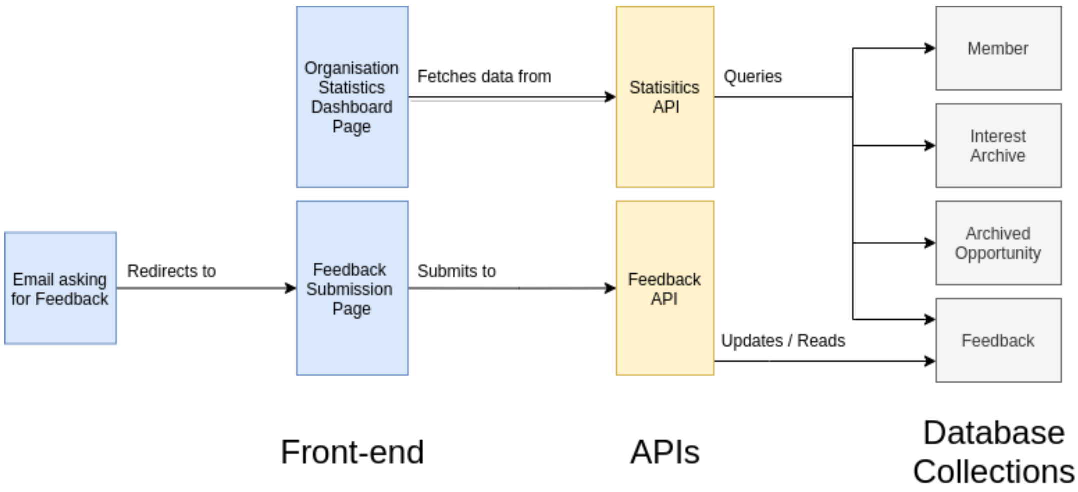

Figure 2: Design of the statistics and feedback system

6.1. Front-end Design

Based on the requirements from the user interviews, (see section 5) the basic MVP for the project would provide an organisation administrator with visibility of their volunteer's total and average hours contributed, contribution by location and contribution by activity type. Based on research from the literature, the contribution would be displayed in a pie graph, as it allows the user to compare categories quickly. The first step in designing the page was the creation of a high-fidelity prototype. This prototype (see figure 3) was built using Figma[5] – an interface design tool. Which was then validated by the design lead for development.

[5] `https://www.figma.com`

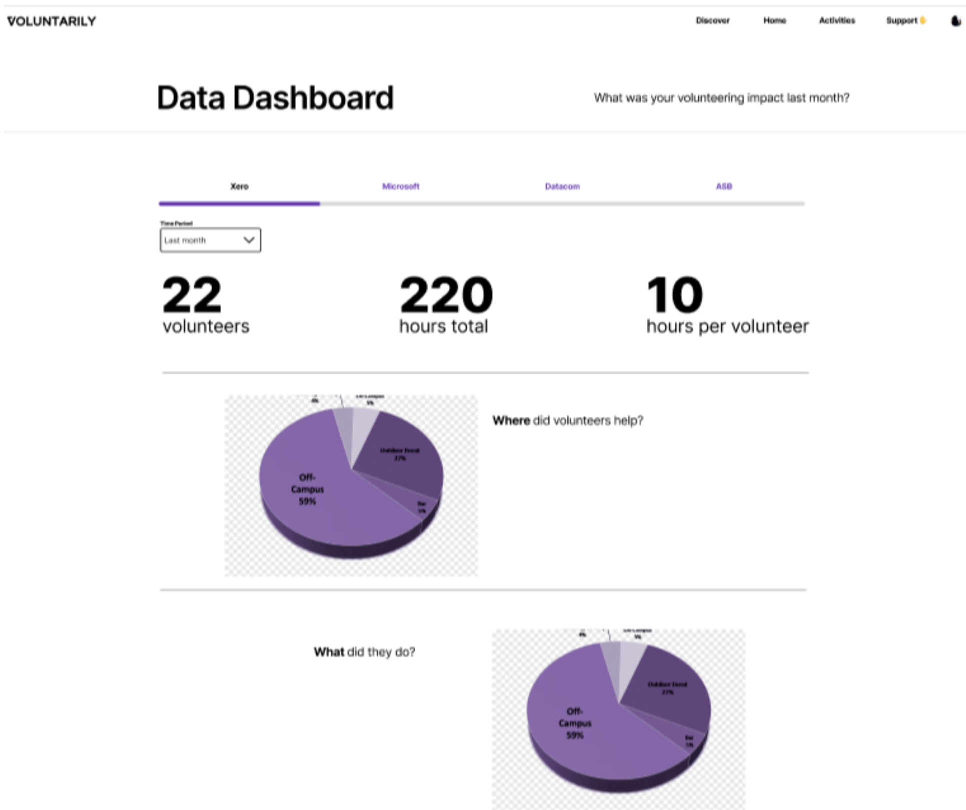

Figure 3: Figma prototype of the statisitics dashboard page

6.2. API Design

6.2.1. Schema

In designing the statistics API, the API schema with the following endpoints was proposed.

- `GET /api/statistics/summary/:orgId/:timeframe` - retrieves the total number of volunteers in the organisation, total hours volunteered and average hours per volunteer.
- `GET /api/statistics/locations/:orgId/:timeframe` - retrieves the number of opportunity attendances by the organisation's volunteers at each location.
- `GET /api/statistics/activitytags/:orgId/:timeframe` - retrieves the weighted occurrences of each activity tag in the opportunities attended.

In designing this API, two approaches were considered. The first approach was the implemented approach described above, where different data types are returned from separate endpoints. The second approach would return all the data (summary, locations, activity tags) from a single endpoint. While the second approach would be slightly more performant as it only requires a single request to be processed, the first approach was chosen for its greater flexibility for future developers, due to the decoupling of endpoints.

The feedback API has been designed to support CRUD actions (supporting create, read, update, delete). This is because the feedback system is expected to be expanded to support more functionality.

6.2.2. Architecture

In implementing both the feedback and statistics APIs, an MVC architecture was used. The front-end components can be described as the ’view’ elements. These components would then make calls to endpoints which were encapsulated in ’controller’ modules. The ’controller’ modules would then perform appropriate updates or queries on the relevant MongoDB models. Using this architecture created a clear separation of concerns, making the system easier to understand and test.

6.2.3. Authorisation

CASL[6] middleware was used for authorisation on the feedback API. CASL is a declarative authorisation library which restricts which resources a user is allowed to access [19]. For example, volunteers could only read their own feedback items, but organisation admins can read feedback belonging to volunteers in their organisation. The main benefit of CASL in this use case was that it allows easy definition and management of business rules for the feedback schema. On the statistics API, CASL was not used as there were issues in dynamically evaluating access to the statistics API resource. Therefore, custom middleware was introduced which verifies the user is an organisation admin for the organisation they are retrieving statistics for.

[6] `https://casl.js.org`

6.3. Query Design

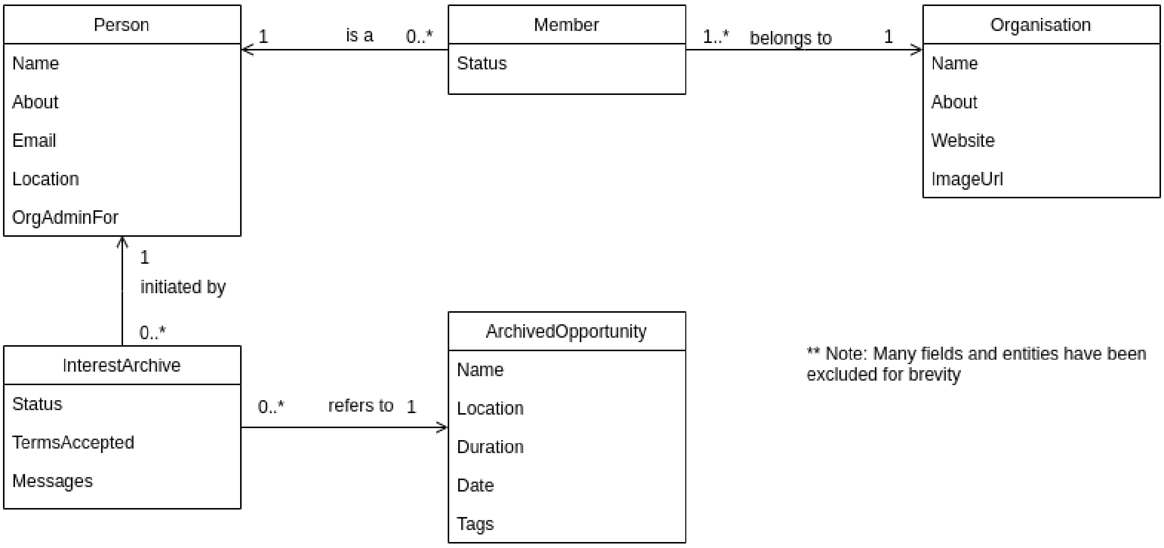

Figure 4: UML diagram of database collections relevant to the query

Significant design consideration was spent in creating an efficient query to retrieve the location, activity tag and volunteer hours data from the existing collections. This query would involve retrieving all attended InterestArchives (with associated ArchivedOpportunity info) for each member of the specified organisation (see figure 4 for relevant entities). The query can be described by the following algorithm (given an organisation ID and a timeframe):

1. Find all members in the specified organisation
2. For each member find all their InterestArchives (where they have shown interest in an opportunity)
3. Filter InterestArchives by status attended (the member must have attended the opportunity)
4. Retrieve the ArchivedOpportunity for each InterestArchive (for location, activity tag and duration info)
5. Filter ArchivedOpportunities to be within the specified timeframe

This general algorithm could be performed either by an aggregation or population approach. Aggregation involves a single complex query to be processed by MongoDB which groups multiple document values. Whereas, the population approach would involve separately querying each collection from Javascript based on the values retrieved from the previous query. Ultimately, while the aggregation approach could be more complex, it was chosen due to it's increased efficiency in terms of database calls. For example, given `n` Members, `m` InterestArchives and `j` ArchivedOpportunities, the population approach would require $n \times m \times j$ database calls, whereas the aggregation approach would only require one database call.

6.4. Database Schema Design

In storing feedback for the feedback system, two approaches were considered. The first approach was to embed the feedback in the InterestArchive document, which indicates that a person attended a particular opportunity. The second approach was to create a new feedback collection. Whilst the InterestArchive would be a logical place to leave feedback; the second approach was chosen because coupling feedback to the InterestArchive schema could prevent feedback being created from other sources that do not require an InterestArchive. Additionally, it would require changing the existing InterestArchive system, which could incur additional bugs.

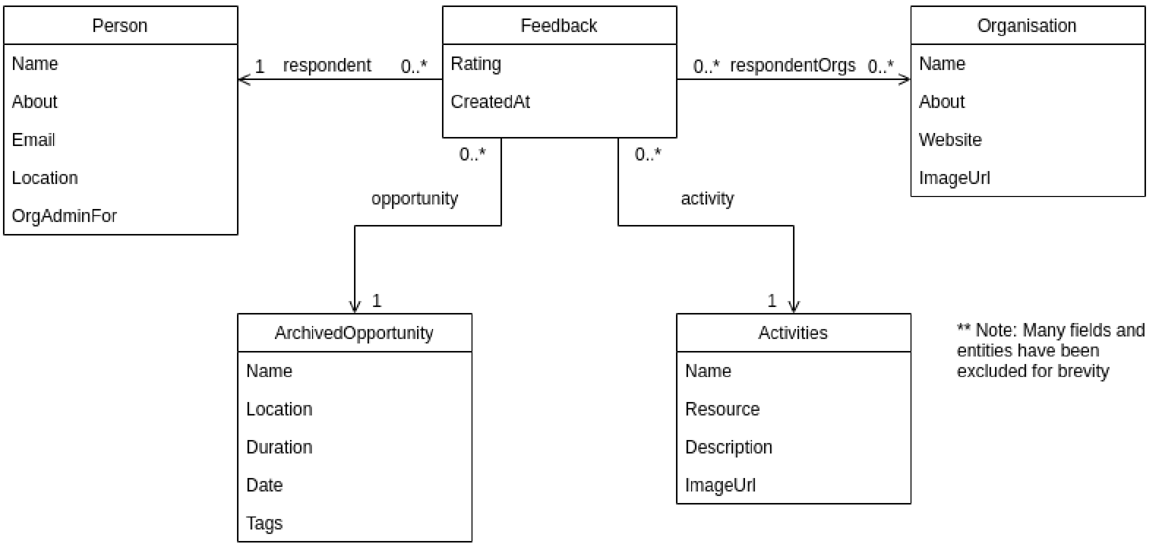

Figure 5: UML diagram of Feedback schema

As seen in figure 5 the fields included within the feedback schema were: *i)* ‘rating’ which indicates the volunteer’s level of satisfaction with the event. *ii)* ‘opportunity’, ‘activity’, which indicates which event the feedback is referring to. *iii)* ‘respondent’ which is necessary to restrict users from only rating the event once. *iv)* ‘respondentOrgs’ which indicates the organisations the respondent belongs to at the time. This field might be considered redundant as this information can be found from an additional join with the person collection using the ‘respondent’ foreign key. However, the schema was designed this way to allow quick retrieval of the feedback from an organisation’s employees without any additional joins. Additionally, when an employee changes or leaves a company the ‘respondentOrgs’ in a feedback object should still remain the same, as the feedback is relevant to companies the employee was in at the time the feedback was given.

7. Implementation

7.1. Volunteering Statistics System

7.1.1. Summary Statistics

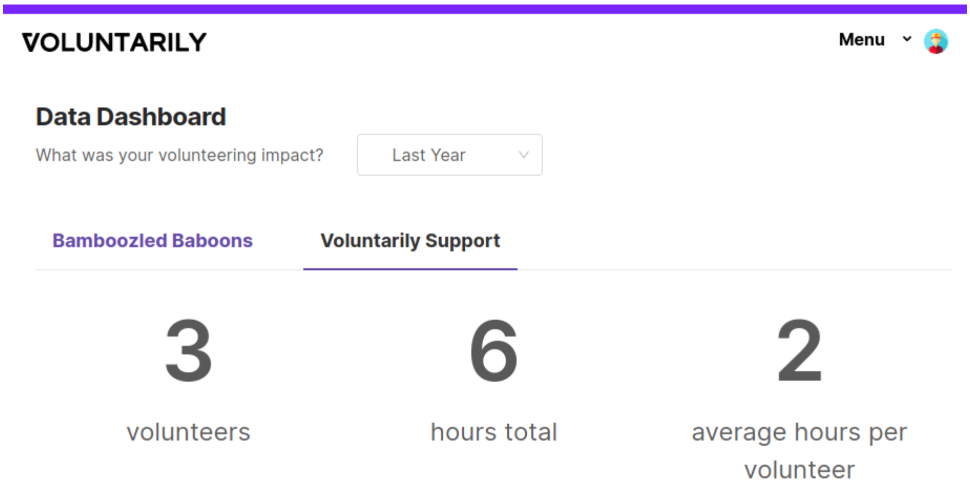

Figure 6: Summary statistics on the organisation statistics dashboard

As an organisation administrator first navigates to the organisation statistics dashboard, the first item they see are the summary statistics (see figure 6.) If the user is an organisation administrator for multiple organisations, they can switch tabs to see statistics for different organisations. Additionally, they can view statistics for different time frames using the dropdown. The number of volunteers, total and average hours data is retrieved by submitting a `GET` request to the `/api/statistics/summary/:orgId/:timeframe` endpoint of the statistics API.

Example use-case:

This feature allows an organisation administrator to assess the volume of the organisation's volunteering efforts. For example, Gill is a CSR manager at a software development company called Sicromoft, located in Auckland. Gill is responsible for the organisation's volunteering efforts and ensuring Sicromoft is adequately supporting their community. The statistic for the number of volunteers allows Gill to judge whether or not enough Sicromoft employees have tried volunteering. The statistics for total and average volunteers also enable Gill to determine whether or not the Sicromoft's volunteers have put enough time into volunteering. From these observations, Gill can then decide whether or not more internal volunteering promotion is needed for Sicromoft to fulfil its corporate social responsibility requirements.

7.1.2. Location Statistics

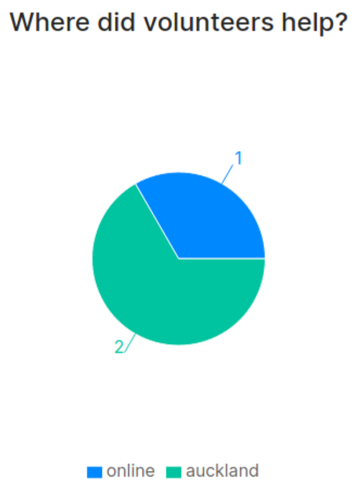

Figure 7: Location statistics on the organisation statistics dashboard

The next item on the dashboard is a pie graph indicating where the organisation's employees have volunteered (see figure 7.) This location data is retrieved by submitting a `GET` request to the `/api/statistics/location/:orgId/:timeframe` endpoint of the statistics API.

Example use-case:

This feature allows an organisation administrator to assess the location diversity of their volunteering efforts. For example, the pie graph allows Gill to quickly compare which locations receive the most or least help from Sicromoft volunteering. It may be the case that Sicromoft has done little to no volunteering in South Auckland, but plenty of volunteering in other areas of Auckland. From this, Gill can determine whether or not some areas are being unfairly prioritised over others due to demographic or other reasons. Following this observation, Gill can help address this bias within Sicromoft by increasing awareness. Hopefully, this will increase the fairness in volunteering opportunities performed by Sicromoft.

7.1.3. *Activity Statistics*

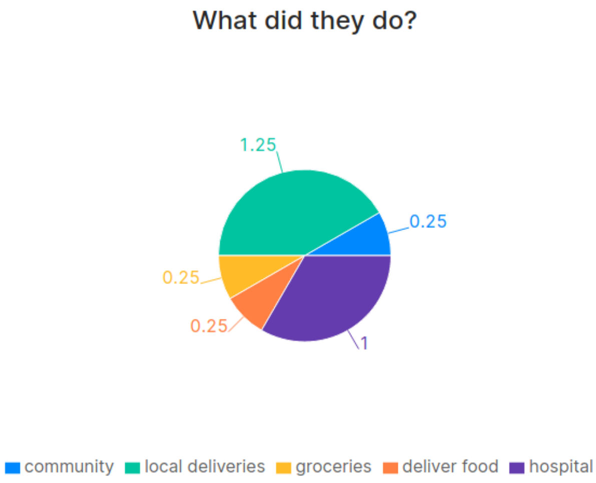

Figure 8: Activity statistics on the organisation statistics dashboard

The next item on the dashboard is a weighted pie graph indicating what type of activities employees have volunteered for (see figure 8.) This activity data is retrieved by submitting a `GET` request to the `/api/statistics/activitytags/:orgId/:timeframe` endpoint of the statistics API. A weighted value is used here to avoid opportunities with many tags skewing the statistics in the pie graph.

Example use-case:

This feature allows an organisation administrator to assess the activity diversity of their volunteering efforts. For example, the pie graph allows Gill to quickly compare which type of activities Sicromoft volunteers are helping with the most. From this, Gill can determine whether or not Sicromoft volunteers need to diversify the activities for which they volunteer. Hopefully, this will increase the activity diversity of Sicromoft volunteering.

7.1.4. *Additional Technologies Used*

In implementing the statistics page, the Recharts JavaScript library[7] was used for creating graphs. Based on research from the literature, Recharts was used over alternatives such as D3.js[8] because Recharts provides a higher-level abstraction and is React-based. D3.js is the most popular charting library; however, it is low-level. While this provides greater control, it has a much greater learning curve. The team determined the

[7] `https://recharts.org`
[8] `https://d3js.org`

time investment required would have been unfavourable, especially in an agile project where requirements may change.

7.2. Feedback Collection System

7.2.1. Feedback Submission

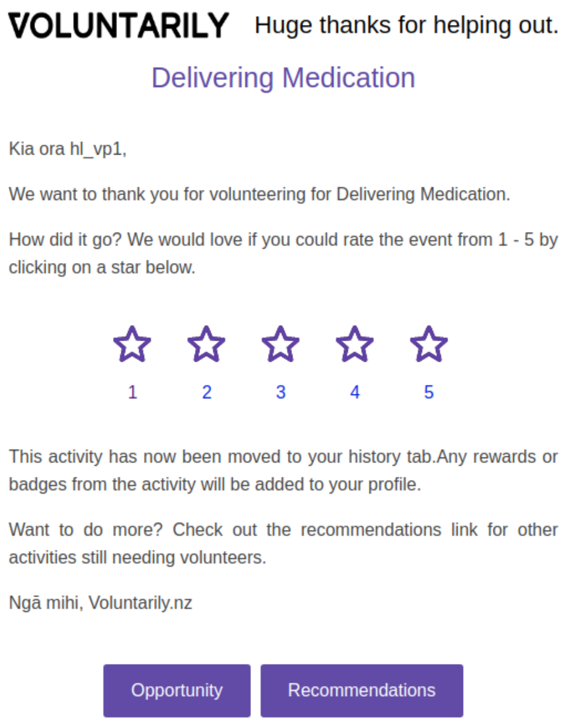

Figure 9: Email received for feedback submission

The first step in implementing the feedback system was first to create some process to get feedback from volunteers. The solution was to email volunteers after they had attended an opportunity (see figure 9.) Inline with web surveying principles explained by Dillman et al. [8] the initial question has been kept as minimal as possible to reduce non-response rate.

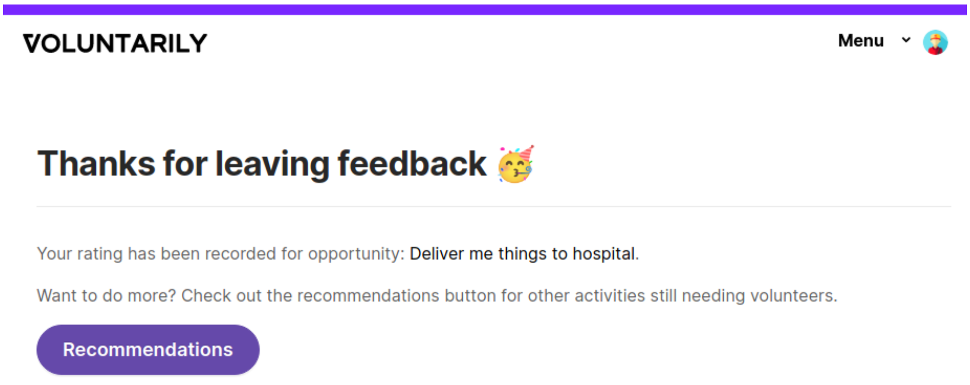

Figure 10: Feedback Submission Page

After selecting a rating on the email, the user is redirected to the feedback submission page, given that they are logged into Voluntarily (see figure 10.) The user's rating response is then recorded through a `POST` request to the `/api/feedback` endpoint. From here, the user can then navigate to view additional recommended opportunities. In the future, the feedback submission page is intended to be extended to have additional questions in which the user can provide more in-depth feedback about the experience and perhaps justify why they gave a certain rating.

Example use-case:

Abe is a software developer at Sicromoft and has used some of his corporate volunteering hours teaching MATLAB at an opportunity administered through Voluntarily. At the event, he noticed it was very disorganised, which led to a lower-quality learning experience for the students. After the event, he receives an email, which allows him to leave an appropriately low rating. This feedback will be recognised by the event organisers who will hopefully get in contact with Abe.

7.2.2. Feedback Statistics

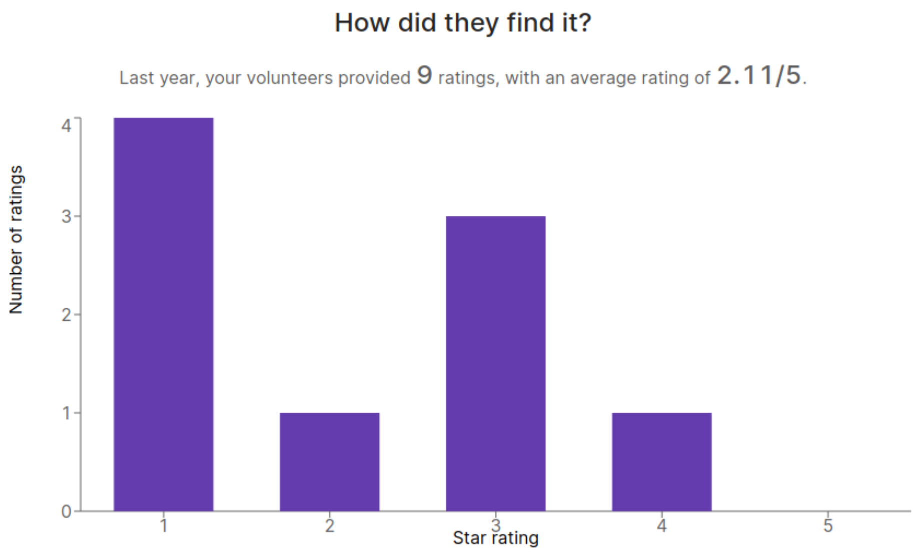

Figure 11: Feedback Rating Statistics on the Organisation Statistics Dashboard

As part of the organisation statistics dashboard, the organisation administrator can see the total number of ratings and the average. Additionally, a bar chart of the feedback ratings is presented. In the future, this information is intended to be accessible by activity providers such that they can see which activities need to be improved.

Example use-case:

Using this feature, Gill, the Sicromoft administrator can judge whether or not the Sicromoft volunteers perceive the volunteering opportunities as a positive experience. If Gill sees many ratings similar to Abe's low rating, Gill can follow up with relevant Sicromoft volunteers to see how volunteering experiences could be improved. Hopefully, this will ultimately result in higher quality volunteering events throughout the platform.

7.2.3. Additional Technologies Used

In implementing the email HTML, the pug.js[9] templating engine was utilised. This provided several powerful features, including the ability to write much simpler and more readable code. Additionally, pug.js makes it easy to write reusable HTML, which allowed easy reuse of existing Voluntarily HTML.

[9] `https://pugjs.org`

8. Testing

For the statistics and feedback systems to be useful to the end-user, it is essential that these systems have the correct behaviour. To verify this, the team utilised automated testing throughout the development of the project in both the front-end and back-end.

8.1. Frontend testing

In implementing the user interface components, unit tests were created for each component addition or change. Having these unit tests in place will make statistics and feedback system more maintainable for multiple reasons. Firstly, these components are likely to be reused, so the components must be robust and functionally correct. Furthermore, if these components need to be changed, developers can easily find if they've broken functionality through the unit tests. These improvements to maintainability are important as a goal of this project was to make a set of components that could be easily reused.

8.2. Backend testing

Throughout the development of the back-end for this project, both unit testing and integration testing were utilised to verify functionality. Unit tests were created for controller endpoints, which would verify the API endpoints would return the correct data given certain database contents. Integration tests would involve running the entire request-response pipeline of the server; ensuring back-end modules would work well together. This was particularly useful when testing the authorisation middleware and role-based functionality. For example, only organisation administrators should be able to access the statistics for their organisation. Ultimately, similar to the front-end testing, the back-end testing suite has improved the maintainability of the systems, which will be useful for future developers.

9. Methodologies

9.1. Project Management

Throughout the development phases of the project, an agile project management methodology was utilised involving KanBan. The work items were managed on a KanBan board facilitated by the project management tool Jira[10]. The work items for the project were divided into user stories, which was beneficial in helping the team understand the purpose of the tasks from the perspective of an end-user. Following the KanBan

[10] `https://www.atlassian.com/software/jira`

methodology, stories were then moved across the board as they progressed (with columns: Todo, In Progress and Done). This practice increased workflow transparency, as Voluntarily team members could easily see the progress on the project. In alternative software development methodologies such as Scrum, it is common to define fixed time periods (known as sprints) in which work is completed. However, due to the part-time nature of the project, the team members decided the more flexible scheduling provided by KanBan was preferable. Overall, the benefits regarding transparency and flexibility resulted in KanBan being an effective methodology for this project.

9.2. Pair programming

An agile software development technique used throughout the project was pair programming. This practice involved one team member sharing their screen through a remote video call. This team member, the driver, would write the code while the other, the navigator, would review each line as it is written. While the immediate productivity may have been lower than individual programming, this practice provided many long-term benefits throughout the project. Firstly, through partner discussions, better designs were obtained, improving long-term maintainability. Additionally, many mistakes were often discovered immediately rather than later, preventing time waste. Furthermore, the partners were able to learn from each other's working styles, allowing each member to make self-improvements. Overall, the benefits regarding long-term efficiency and learning have proven Pair programming to be an effective technique for this project.

9.3. Development Workflow

The Voluntarily team used a trunk-based development workflow called the "shared repository" model. All development of a feature or bug occurs in its own feature branch (e.g. `feature/ratings` or `bug/cant-rate`). After the feature is completed, pull requests are made to the `master` branch. The continuous integration pipeline would then check all newly introduced code. These checks ensure linting and formatting rules have been followed, the front-end and back-end can build, and all unit tests pass. Additionally, other team members perform a code review on the changes, pointing out any issues through manual inspection and testing it on their own machine. This workflow has prevented many errors, inconsistencies, and design flaws and has ultimately ensured all code introduced in this project has been of high quality.

10. Discussion

10.1. Research Intent

As discussed in section 4, the research intent of this project was to identify and implement methods to report volunteering impact and success metrics through the Voluntarily platform. Through rigorous design and effective development methodologies, the team has exceeded the requirements for an MVP of such a system as defined by the Voluntarily team. The impact and success metrics in the current system, while simple, are suitable for the platform at its current stage. Furthermore, another project goal was to implement an extensible system. This goal has also been achieved through a set of reusable, robust and thoroughly tested components and examples from which future contributors can extend the system.

From an academic context, the implementation has proven that simple survey design and data visualisation techniques were the most valuable research during the creation of the MVP. Perhaps, volunteering impact metrics on schools could have been explored further. Additionally, data mining research was not explored further throughout this project. Ultimately, due to a slight misalignment of the Voluntarily requirements and the research domains, the scope of research exploration was reduced (discussed further in section 10.3.)

10.2. Technologies

As explained in section 2, Voluntarily uses the MERN stack of technologies, this presented several opportunities and challenges throughout the project. MongoDB allows unstructured data which resulted in many documents from a data backup having inconsistent data representation. This often resulted in confusing results when testing queries. However, unstructured data allows simpler iterative development as data migrations are not required. Node.js was advantageous as it allows the "JavaScript everywhere" paradigm [20], thus there is a reduced learning curve when contributing to both the front-end and back-end. However, since JavaScript is dynamically-typed, there was an increased chance of bugs due to type errors. Express proved to be advantageous as it provides utilities to build web APIs quickly. For example, many low-level complexities of HTTP request handling such as parsing cookies, and payloads are abstracted. React.js was advantageous mainly because of its component-based nature, which allowed easy reuse and testing of UI components. Overall, the modern technology stack used by Voluntarily was beneficial in terms of iterative development, ease of learning and testability, making it highly effective for this project.

10.3. Agile and Start-Up Environment

Voluntarily is a start-up that uses an Agile software development methodology which has presented benefits throughout this project. Iterative development allowed frequent consultations with the product lead and quick integration of feedback. However, this agile environment also presented several unique challenges. The original research did not perfectly align with new requirements gained from user interviews. For example, data mining was not necessary as part of the MVP. Additionally, as part of the research contribution, the project had intended to explore effective volunteering impact metrics for schools. However, the requirements only provide metrics relevant to volunteer providers, as schools were considered less important stakeholders. Furthermore, due to the impact of COVID-19, new requirements have resulted in a large pivot of the platform (e.g. a new ask, offer system was introduced). This pivot has resulted in legacy code, unnecessarily complex areas of code, and reworking of application concepts.

11. Conclusions & Future Directions

This project has laid the foundation for Voluntarily's reporting system, which will hopefully help the continuity of the Voluntarily project. The implemented statistics system will help organisations assess the impact and success of their volunteering efforts. Furthermore, the feedback system will help drive higher quality volunteering events through Voluntarily. The requirements from the user interviews have driven the project in a slightly different direction than was envisioned from the literature review. But through considered design and effective development methodologies, the team has implemented a system which satisfies the MVP requirements, which can be easily extended by future contributors.

There are still many areas of the project that could be extended. As discussed in section 10.3, the statistics system only includes metrics relevant to volunteer providers. Future work could extend this system to enable other administrators to access different metrics that make sense to them. Mentioned in section 7.2.2, activity provider administrators could see ratings for activities that their organisation has created. From here, they can see which activities need improvement. As explained in section 10.3, additional metrics regarding the impact on schools could be added to measure the educational impact of volunteering events. As mentioned in section 7.2.1, the feedback submission page could be extended with more questions. Perhaps questions regarding MPS (see section 3.1) could be utilised to find potential activity improvements. Adding these features would provide more users with greater visibility of volunteering efforts, and could ultimately drive more people to use Voluntarily.

Acknowledgements

The author would like to thank their project supervisor, Andrew Meads, and their project partner, Jordan Sim-Smith, for their continual guidance and contribution throughout this project. The author would also like to thank the Voluntarily team, especially Andrew Watkins and Walter Lim, in helping them understand the project requirements and for creating the platform to make the project possible.

References

[1] S. Sekar and L. Dyaram, "What drives employees to participate in corporate volunteering programs?" *Social Responsibility Journal*, 2017.

[2] "Technology," [Accessed September 17th 2020]. [Online]. Available: https://nzcurriculum.tki.org.nz/The-New-Zealand-Curriculum/Technology

[3] V. Millette and M. Gagné, "Designing volunteers tasks to maximize motivation, satisfaction and performance: The impact of job characteristics on volunteer engagement," *Motivation and Emotion*, vol. 32, no. 1, pp. 11–22, 2008.

[4] H. J. Richard and G. Oldham, "Motivation through the design of work: Test of a theory," *Organizational behavior and human performance*, vol. 16, no. 2), 1976.

[5] S. K. Kane and P. V. Klasnja, "Supporting volunteer activities with mobile social software," in *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, 2009, pp. 4567–4572.

[6] U. D. of Health, H. Services *et al.*, "Successful strategies for recruiting, training, and utilizing volunteers: A guide for faith-and community-based service providers," *Substance Abuse and Mental Health Services Administration, Center for Substance Abuse Treatment*, 2005.

[7] R. M. Groves, *Survey errors and survey costs*. John Wiley & Sons, 2005, vol. 581.

[8] D. A. Dillman, R. D. Tortora, and D. Bowker, "Principles for constructing web surveys," in *Joint Meetings of the American Statistical Association*, vol. 64, 1998, pp. 1–16.

[9] D. Behr, K. Meitinger, M. Braun, and L. Kaczmirek, "Web probing-implementing probing techniques from cognitive interviewing in web surveys with the goal to assess the validity of survey questions (version 1.0)," 2017.

[10] K. Meitinger and D. Behr, "Comparing cognitive interviewing and online probing: Do they find similar results?" *Field Methods*, vol. 28, no. 4, pp. 363–380, 2016.

[11] M.-S. Chen, J. Han, and P. S. Yu, "Data mining: an overview from a database perspective," *IEEE Transactions on Knowledge and data Engineering*, vol. 8, no. 6, pp. 866–883, 1996.

[12] J. Jackson, "Data mining; a conceptual overview," *Communications of the Association for Information Systems*, vol. 8, no. 1, p. 19, 2002.

[13] M. Rowe, "Mining user lifecycles from online community platforms and their application to churn prediction," in *2013 IEEE 13th International Conference on Data Mining*. IEEE, 2013, pp. 637–646.

[14] W. H. Inmon, "The data warehouse and data mining," *Communications of the ACM*, vol. 39, no. 11, pp. 49–51, 1996.

[15] S. Chaudhuri and U. Dayal, "An overview of data warehousing and olap technology," *ACM Sigmod record*, vol. 26, no. 1, pp. 65–74, 1997.

[16] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[17] S. S. Ajibade and A. Adediran, "An overview of big data visualization techniques in data mining," *Int J Comput Sci Inf Technol Res*, vol. 4, no. 3, pp. 105–113, 2016.

[18] Saring, "11 javascript data visualization libraries for 2019," 2019. [Online]. Available: https://blog.bitsrc.io/11-javascript-charts-and-data-visualization-libraries-for-2018-f01a283a5727

[19] "Casl. isomorphic authorization javascript library," 2020, [Accessed August 1st 2020]. [Online]. Available: https://casl.js.org/v4/en

[20] "About node.js," 2020, [Accessed August 1st 2020]. [Online]. Available: https://nodejs.org/en/about/