Department of Electrical, Computer, and Software Engineering

Part IV Research Project

Final Report

Project Number: 69

Measuring Volunteering Success for Voluntarily

Jordan Sim-Smith

Harrison Leach

Andrew Meads

30/10/2020

Declaration of Originality

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

Name: Jordan Sim-Smith

ABSTRACT: Voluntarily is an open-source volunteering platform that aims to connect schools with corporate volunteers who can teach digital technology content. Clients of Voluntarily are interested in measuring the success and impact of this platform. This project consists of both research and development components. The literature is examined with a focus to four key areas; volunteer motivation, data collection, data mining and data visualisation. Applicable concepts from the literature are discussed and incorporated into the development of the project. A data reporting dashboard has been created, featuring volunteer attendance statistics. A feedback collection mechanism has been constructed, asking volunteers to rate their experience after each opportunity attended. User interface prototyping and query design satisfied Voluntarily's non-functional requirements of performance, aesthetic consistency and project maintainability. The Agile software development methodology and pair programming practice were followed to produce a high-quality implementation under fixed time and resources constraints. The implementation provided is highly extensible, providing a foundation for future development to enhance the platform's reporting capabilities.

1. Introduction

Voluntarily is an open-source volunteering platform that aims to connect New Zealand (NZ) schools with corporate volunteers who can teach digital technology content. Recent changes to the NZ education curriculum mandate the teaching of technology skills in the classroom. However, teachers tend to lack the necessary experience to teach in this area and would benefit from expert assistance. Fundamentally, the Voluntarily platform facilitates matchmaking between parties offering and asking for help.

Clients of voluntarily are interested in measuring the impact and usage of volunteers through the platform. As such, there is an opportunity to develop a data gathering and reporting module (data module) for Voluntarily. This data module will communicate the successes and challenges of volunteering activities through the platform, providing a foundation for improvement.

This report documents the research, design and development of the data module. A brief literature review is included, with focus on volunteer motivation, data collection, data mining and data visualisation techniques. An overview of the features implemented on the Voluntarily platform is provided. Subsequently, the software design, testing and methodologies contributing to this implementation are discussed. Finally, the project goals

and technologies chosen are reflected upon, and future work for the project is outlined.

2. Voluntarily

Voluntarily is a non-profit initiative funded by the Pam Fergusson Charitable Trust (PFCT). The PFCT is committed to creating educational opportunities for school children with a focus on technology and innovation [1].

Recent changes to the NZ education curriculum have mandated the teaching of technology skills in the classroom. Unfortunately, teachers tend to lack the necessary experience to teach in this area, and would benefit from expert assistance. Voluntarily aims to overcome this problem through matchmaking between parties offering and asking for volunteering help. Voluntarily aims to streamline this process to encourage volunteering, and to help build long-lasting relationships between these parties.

For example, a secondary school teacher may ask for help with a robotics activity on the Voluntarily platform. Later, a robotics engineer may view this opportunity on the platform and offer their time and expertise to help teach this activity. Another organisation may provide the details and equipment required to facilitate this activity. This three-way matchmaking process between teachers, volunteers and content providers forms the core of the Voluntarily platform.

Corporate Volunteering (CPV) is the act of giving employee time, knowledge or skill as part of community service or outreach, often sponsored by the company [2]. CPV programs are offered by several large technology companies in New Zealand, including Datacom and Xero [3, 4]. Voluntarily aims to connect with these companies to encourage employees to utilise their CPV hours for classroom volunteering.

Voluntarily is an open-source software (OSS) product, mainly relying on development efforts from the community. The product is a full-stack web application, developed using modern technologies to encourage open source collaboration. The architecture of the product utilises the MongoDB, Express, React and Node (MERN) software stack. The entire MERN stack is written using JavaScript (JS), providing a minimal barrier to entry for new contributors. Also, third party libraries such as Redux[1] for global state management and CASL[2] for

[1] https://github.com/reduxjs/redux
[2] https://github.com/stalniy/casl

role-based authentication are used.

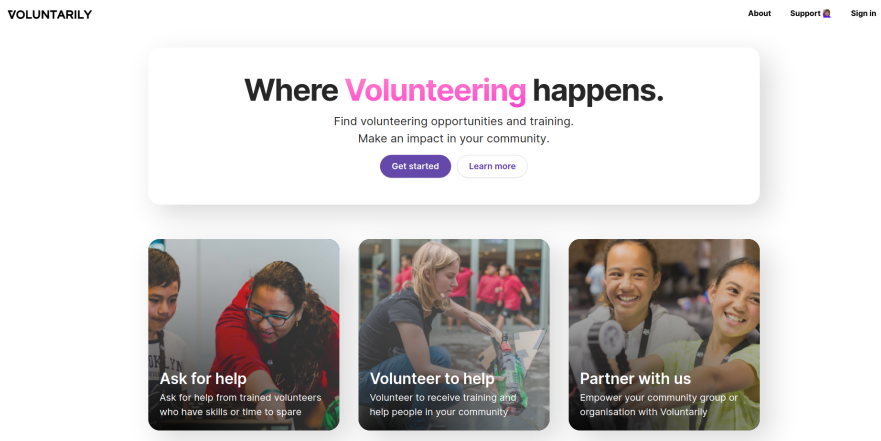

Figure 1: Home page of the Voluntarily application.

3. Research Intent

Organisation administrators (OA) are Voluntarily client representatives that are responsible for managing the usage of the platform within their organisation. User interviews with potential clients have indicated that OAs are interested in measuring the usage and success of the platform. Specifically, the user interviews mentioned an interest in knowing how many volunteers have participated in volunteering events in the last month or year, and the total number of hours spent volunteering. These requirements have formed the basis for the Minimum Viable Product (MVP) for this research project.

The purpose of the research project is to develop a data module for the Voluntarily platform, capable of measuring the success of volunteering activities. An acceptable implementation will measure volunteering usage according to the elicted user requirements, and visualise the information appropriately so that it is easily understandable by Voluntarily users.

The project consists of two main components; research and development. Research is concerned with analysing the literature for background, concepts, techniques and suitable approaches for the project's elements. Development is concerned with leveraging the appropriate research findings and providing an implementation that is fit for purpose. Early stages of the project may be more research-oriented, and later stages may be more development-oriented. However, each component feeds into the other; both are necessary at all stages of the project.

Research conducted in the early stages of the project is structured around the general steps in evaluating the success of a volunteering program [5].

1. Corporate volunteering and employee motivation, to identify success from the volunteer perspective;
2. Survey and data collection techniques, to gather data from Voluntarily users;
3. Data mining concepts, to analyse collected data;
4. Data visualisation techniques, to display analysis results in a meaningful format;

4. Literature Review

4.1. Volunteer Success

Volunteering success, as defined by Voluntarily, is multifaceted. Educational impact on the students, retention of volunteers and satisfaction of teachers all contribute to volunteering success. This research project is focused on measuring success from the volunteer perspective. Techniques used need to be comprehensively supported by the literature and achievable with the current state of the Voluntarily platform.

According to Voluntarily, volunteer success and retention are derived from volunteer satisfaction from contributing to an activity, and their overall motivation.

4.1.1. Design and management of CPV programs

A significant factor of employee volunteer motivation is the design and management of CPV programs [2]. Programs that have a clear purpose, and are well designed to serve that purpose tend to result in higher volunteer involvement. Volunteers prefer to participate in projects that are efficiently run and have clear task assignments [6].

Another factor identified in the literature is that volunteers prefer to receive information in manageable quantities [7]. CPV programs that overload their volunteers with newsletters or invitations are likely to have lower volunteer motivation.

4.1.2. Self-oriented vs other-oriented motives

A distinction is made in the literature between self-oriented motives and other-oriented motives [8]. Self-oriented motives refer to the fulfilment of personal needs and wants, including increased self-esteem, skill development, building relationships and career growth. Other-oriented motives refer to the increased wellbeing of others in the community. Self-oriented motives do not significantly impact volunteer motivation, while other-oriented motives do [2]. Thus, other-oriented motives should be used over self-oriented motives to determine volunteer commitment, involvement and success of CPV activities.

4.1.3. Volunteer task characteristics

The characteristics of volunteering tasks affect volunteer motivation [7]. Volunteers appreciate the freedom to choose activities that they are passionate about, where they can develop new skills and gain relevant experience. Volunteer motivation is also increased through minimising time and distance costs. High overhead costs of participating in volunteering activities can discourage volunteers, who are often struggling to find time between other commitments.

A significant characteristic of volunteer tasks impacting motivation is the dispersal of frequent and iterative feedback [2, 7]. Volunteers are interested in feedback about the impact of their volunteering work [7]. Volunteers tend to fondly remember instances where they felt they made a positive impact on a person's life [7].

Iterative feedback helps volunteers recognise the impacts of their efforts on the community [2].

4.1.4. Measuring task characteristics

Hackman and Oldman describe a method for task characteristic collection and quantitative analysis [9]. Skill variety, task identity (working towards a distinct goal), task significance, autonomy (independence, freedom) and feedback were defined as key task characteristics for the method.

Hackman and Oldman define a job diagnostic survey used to quantitatively measure these characteristics [9]. Each question was measured between 1 (very accurate) and 7 (very inaccurate) on a likert scale. Example questions were provided, such as:

- "My volunteer work requires me to use a number of complex or high level skills" (skill variety);
- "My task is arranged so that I do not have the chance to do an entire identifiable task from beginning to end" (task identifiability, reversed);
- "This volunteer job is one where a lot of other people can be affected by how well the work gets done" (task significance);
- "This volunteer job denies me any chance to use my personal initiative or judgment in carrying out the work" (autonomy, reversed);
- "The volunteer work itself provides very few cues about whether or not I am performing well" (feedback, reversed);

Hackman and Oldman propose the Motivation Potential Score (MPS) to aggregate the task characteristics [9]. It is proposed that the MPS aggregate is more useful than each metric individually. Jobs with high MPS tend to have higher internal work motivation and performance.

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

4.2. Data Collection

Data collection is an essential aspect of evaluating a volunteer program [5]. The literature describes surveying techniques to improve the quality of survey results. The construction of web surveys can be optimised to reduce non-response [10].

4.2.1. Respondent friendly survey design

The effectiveness of surveys tends to depend on overcoming four sources of error [11]. Coverage error where some units in the population may have no chance of selection and are not represented in the sample. Sampling error which is the result of only surveying a fraction of the population. Measurement error where poor questions and interviewing lead to inaccurate answers. Nonresponse error where non-responsive participants have the

potential to impact survey results.

Surveys should be designed such that each participant has an equal chance of responding with accurate answers to minimise measurement and nonresponse error [10]. Participants must be able to comprehend what is expected of them, what actions are required to respond, and be motivated to make those actions.

The web survey should be introduced with a welcome screen that is motivational and instructs respondents on the immediate action required [10]. Respondents should be presented with immediate feedback that they have followed the correct link. The welcome screen should be designed to introduce the survey as simply as possible, avoiding information overload.

The first survey question should be easy to comprehend and answer by all respondents [10]. It should be interesting as it will be judged by the user as to whether it is worth proceeding with the survey. The first question should not ask for respondent background or demographics [12].

Survey design should conform to standards and popular design [10]. Respondents tend to draw from past experiences when completing a survey, so actions must follow their intuition. Intuitive design may allow users to immediately recall how to switch between inputs or submit the survey from their past experiences.

4.3. Data Mining

Data mining extends on traditional data analysis, incorporating techniques such as numerical analysis, pattern matching and other areas of artificial intelligence [13]. Data mining involves the extraction of implicit, previously unknown and high-level information from various data sources [14].

4.3.1. Types of data mining discoveries

There are six main types of data mining that are used with databases [13, 14]:

1. Association rules—these are significant associations among items such that the presence of some items implies the presence of other items [13, 14]. Generally, association rules follow the pattern—if an event X happens, an event Y usually happens as well. Association rules are commonly sought in transactional and relational databases. Data mining for association rules tends to be computationally expensive.

2. Data summarisation—this provides a high-level overview of database contents [13, 14]. Data found in databases are often detailed and primitive. It is often useful to summarise large sets of raw data to present it at a higher level.

3. Classification—this is the process of finding similarities between database objects and classifying them using a classification model [13, 14]. Classification requires a training set of classified objects that contain the same attributes as the test set.

4. Clustering—this is the process of grouping objects into classes of similar objects [13, 14]. Clustering differs from classification as the classes are not pre specified, instead they are generated as part of the clustering algorithm. Clusters identify densely populated regions according to some distance measurement, across multiple dimensions.

5. Regression—this uses the relation between two or more quantitative variables such that the dependent variable can be predicted from the other, independent variables [13].

6. Prediction—this is similar to classification, except classes are not qualitative, discrete attributes, but rather continuous attributes [13]. The goal of prediction is to find the numerical value of a target attribute for unseen objects.

4.4. Data Visualisation

Data visualisation involves the presentation of data in a graphical or pictorial layout. It is intended to help decision-makers to understand complex ideas at a high level [15]. Interactivity in data visualisation provides more data to the viewer, allowing for deeper understandings of the data characteristics and patterns [15].

4.4.1. Visualisation methods

Line charts display the relationships between each variable in the chart [15]. They are frequently used to make comparisons between multiple variables at the same time. Multi-comparison is facilitated through the stacking of lines on the chart. Line charts can also be useful to display the change of a variable with regard to another variable.

Pie charts provide a clear visual comparison of quantities through the various sizes of the slices [15]. Pie charts are effective when representing fewer components (slices). Each slice should be a percentage of a whole.

Scatter plots provide a two-dimensional data representation using Cartesian coordinates [15]. The relationship between the two variables is represented using vertical and horizontal distance. Scatter plots effectively display the scatter or spread of the data, as well as how strongly two variables are related.

4.4.2. Data visualisation JavaScript libraries

D3.js is a popular and extensive JS data visualisation library, following modern web standards and utilising HTML, CSS and SVG [16]. ChartJS leverages the HTML5 canvas element to provide customisable charts and animations. ThreeJS is built for creating 3D animations using WebGL, and is useful for visualising data in 2 or 3 dimensions. Recharts a lightweight library built on top of D3.js with React, providing customisability and declarative React components.

5. Implementation

The implementation of the data module consisted of two main components; the reporting dashboard and the feedback collection mechanism. Clients of Voluntarily are interested in basic reporting for volunteering activity. The implementation of the reporting dashboard, the minimum viable product (MVP) of the data module, satisfies these user requirements. The feedback collection mechanism was developed as an extension to the MVP, with integrations to the dashboard.

5.1. Reporting Dashboard

The initial user interviews conducted with Voluntarily OAs indicated that OAs are interested in measuring the usage and impact of the platform on the organisation's volunteering effort. OAs mentioned that important measurements were the number of volunteers, the total number of hours spent volunteering and the average volunteering time per volunteer.

The reporting dashboard was implemented as a page on the Voluntarily web application that is only visible to authenticated OAs. Organisations under a user's administration are displayed as tabs, allowing users to quickly select which organisation they would like to view reports for. A simple dropdown allows the user to select a

timeframe to filter the data presented. Currently, supported timeframes include last month and last year.

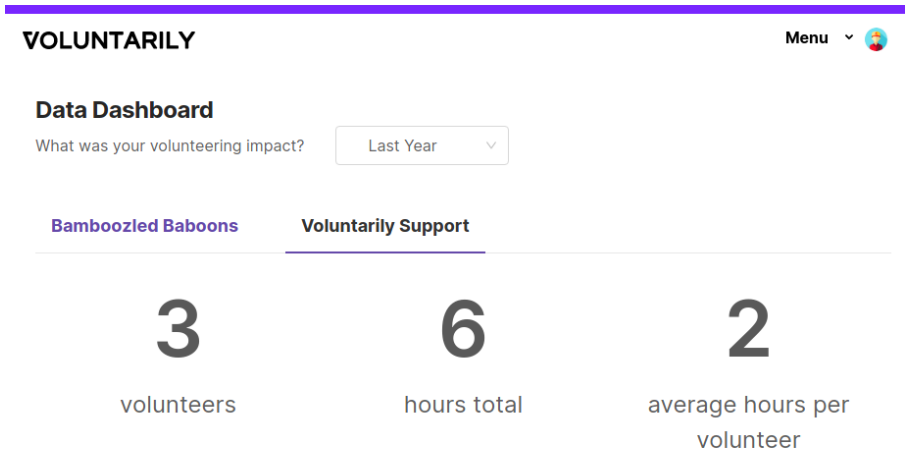

Figure 2: The summary report on the reporting dashboard.

The MVP of the reporting dashboard required the summary information to be queried and displayed. An API endpoint was exposed on the server, calculating the number of volunteers, the number of hours spent volunteering, and the average hours spent volunteering from using database documents. The Voluntarily client application performs AJAX requests for this data on page load.

An essential requirement from the Voluntarily team was that the reporting dashboard design was highly modular. The current dashboard design and displayed information are tailored towards OAs. However, in the future, the dashboard may be used by a wide range of users, each interested in varying reports. As such, the data dashboard was designed such that reporting components (summaries, graphs) can be added and removed from the page easily. This was achieved by creating each reporting React component with a single responsibility, to minimise dependencies between components. Similarly, each reporting component is responsible for retrieving and storing its data to display.

As a minor extension to the MVP, pie charts were included to show the distribution of the locations and nature of volunteering events attended. To facilitate this, the database query was extended to include activity tags and location data. Similar to the summary information, API endpoints were exposed on the server. The Recharts[3] React charting library was used to render the requested location and activity data as pie charts.

[3] https://github.com/recharts/recharts

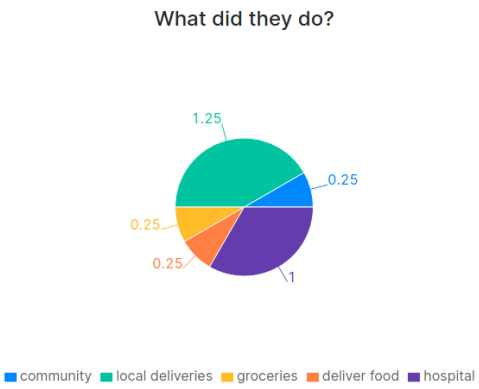

Figure 3: The activity type pie graph on the reporting dashboard.

5.2. Feedback Collection

The requirements for the MVP were relatively simple, and the development of the reporting dashboard was completed relatively quickly. A significant focus area of the research conducted involved measuring volunteering success. Following discussions with the Voluntarily team, it was decided that a feedback collection mechanism should be developed, and volunteer feedback should be included on the reporting dashboard.

The first component involved in the feedback collection implementation was the construction of a RESTful API to manage feedback entities. A new mongoose[4] schema was created, and a set of Express[5] endpoints were developed to expose CRUD operations on the feedback collection. Authentication and authorisation were provided using the CASL framework and Express middleware.

The next component developed was a new page on the Voluntarily application, which would create a feedback entity using the feedback API, when visited. The page records feedback automatically, without user interaction, to streamline the submission process to a single click.

After attending a volunteering opportunity, volunteers are sent an email, thanking them for their commitment. The email template was modified to include star ratings, each linked to the feedback submission page. For example, the volunteer could select the first star and would be directed to the feedback submission page, with a rating of 1.

[4] https://github.com/Automattic/mongoose
[5] https://github.com/expressjs/express

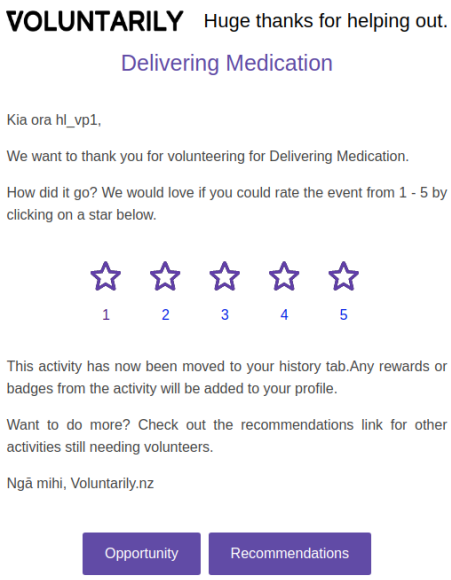

Figure 4: The modified email template sent to volunteers after attending an opportunity.

Lastly, the reporting dashboard was extended to query feedback submitted by employees of the selected organisation, and to display the results in a bar chart. At a glance, OAs can see the average rating and distribution of ratings provided by their volunteers.

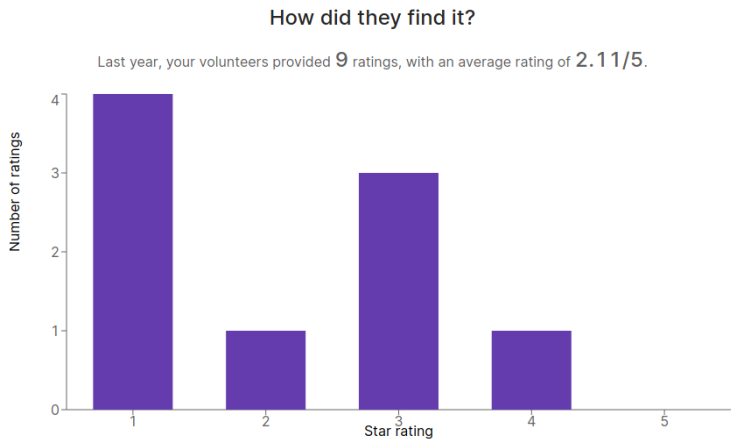

Figure 5: The feedback rating average and bar chart on the reporting dashboard.

5.3. Example Use Cases

A persona analysis is used to illustrate potential user flows through the data module. Two personas, Alice and Bob, are used to describe an example user flow through the system.

Alice is an employee of Xero and an OA for the Voluntarily platform. As an OA, Alice manages CPV efforts from Xero employees. Alice is interested in the amount of volunteering work that is completed through the Voluntarily platform, in terms of hours and the number of participants.

Alice can visit the newly implemented reporting dashboard on the Voluntarily application to find this information. Alice can choose to view the previous month's volunteering effort, or the previous year's, using the timeframe dropdown. The report summary provides statistics on the total number of hours spent volunteering, the number of volunteers participating and the average time spent per volunteer.

Bob is also an employee of Xero and a volunteer on the Voluntarily platform. Bob uses Voluntarily to find opportunities where he can volunteer his time and teach robotics to school children. Opportunity discovery, matchmaking and attendance functionalities are already implemented and will be omitted from the user flow description.

After attending a robotics teaching opportunity, Bob receives an email, thanking him for his commitment. This email contains buttons to submit a rating, out of 5 stars. Bob found the opportunity to be well organised and selected a 5-star rating. The newly implemented feedback submission page on the Voluntarily application is opened, thanking Bob for his feedback.

Later, Alice views the reporting dashboard for a second time. Bob's feedback is anonymously reported in the feedback average rating and bar chart component. Alice can use the summary and feedback data reported to evaluate the success of Xero's CPV program and learn where improvements can be made.

6. Design

The implementation of the data module was developed on the existing Voluntarily platform. Any modifications made by the team for this research project would impact future Voluntarily users and developers. As such, the design of the data module needed to satisfy Voluntarily's essential nonfunctional requirements. Namely, performance, aesthetic consistency and maintainability.

A three-tier, layered architecture was used to promote maintainability through the separation of concerns (see

Figure 6).

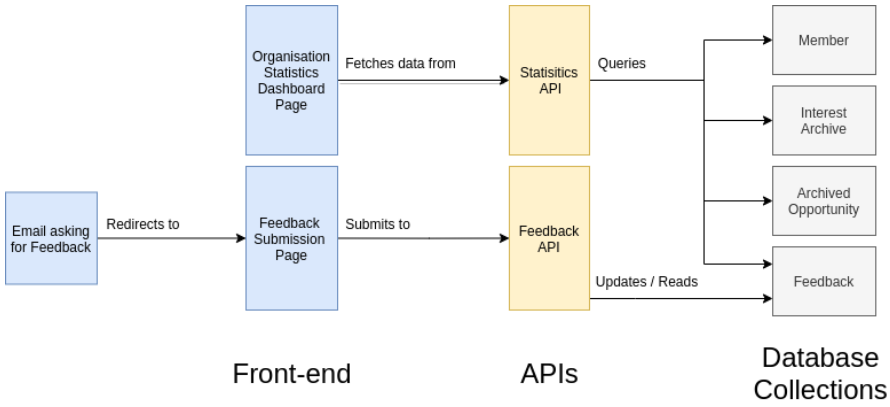

Figure 6: Architectural overview of the data module.

6.1. User Interface Design

A large proportion of the data module's implementation was user-facing. This included the reporting dashboard web page, the feedback submission web page and the post-event gratitude email template. It was important that the user interface (UI) design of these components were consistent with the overall design of the Voluntarily application. Consistent design provides familiarity between pages and allows users to learn to use the application faster [17]. Consistency also promotes reusability, saving development and maintenance costs.

Initial stages of the UI design process involved research into similar, existing components on the Voluntarily platform. This step allowed the team to develop a good understanding of existing design patterns and components.

Following the research stage, UI mockups were designed using Figma[6]. Figma is a web-based prototyping tool that can be used for UI design. A mockup of the reporting dashboard was quickly designed using shared components from the Voluntarily Figma library. This rapid prototyping aimed to attain feedback from the Voluntarily lead designer as early as possible. Feedback incorporated early, in the design phase is much cheaper than when incorporated later, in the development phase.

When implementing the data module's web pages, the Ant Design[7] UI component library was used to remain consistent with the rest of the application. Ant Design is an enterprise-level UI design language and React com-

[6] https://www.figma.com/
[7] https://github.com/ant-design/ant-design

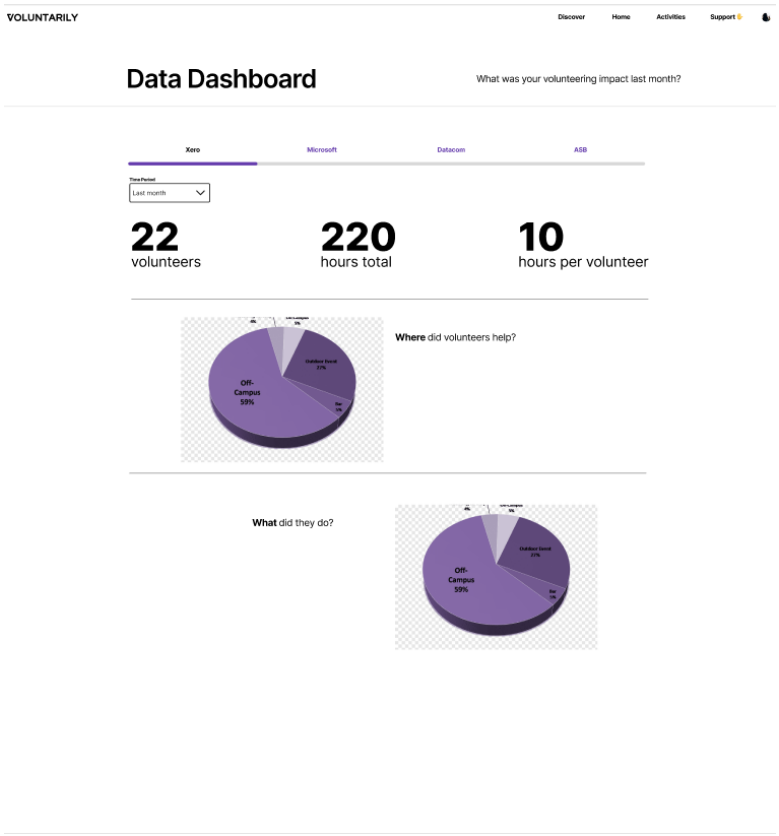

Figure 7: Figma prototype of the reporting dashboard.

ponent library. Ant Design components, such as Buttons, Inputs and Headings, allowed for rapid development of these pages. Also, the styling for these components was inherited from Voluntarily's global theme, meaning that the pages could be made aesthetically consistent, easily.

6.2. Database Query Design

A significant design component of the data module was constructing the MongoDB query to retrieve volunteer attendance statistics. This query would be executed for each API request so that the information reported was always accurate. The query needed to be performant, as multiple reporting components would use it on the reporting dashboard.

The query construction was problematic, as multiple collections needed to be efficiently traversed to attain the final result. In plain English, the query was to retrieve the opportunity details for all attendances by volunteers from an organisation. The arrangement of document schemata and direction of references meant that the query would have to traverse the Member, InterestArchives and ArchivedOpportunities collections while filtering on

the opportunity date and member organisation.

The standard mongoose method of attaining document data from a document reference is to use the populate method. Each call to populate requires an additional database query to be executed, to retrieve that data. Each step of the query had the potential to produce a large intermediate result set. Populating references in each document in these intermediary result sets could result in an enormous number of network calls to the database, significantly reducing the performance of the query.

The solution to optimising this query's performance utilised MongoDB's aggregation functionality. Here, documents are processed through a multi-stage pipeline that transforms the documents into an aggregated result [18]. Filters, transformations and reference lookups could all be done in a single query, minimising the number of network calls between the application server and the database.

7. Testing

The Voluntarily codebase is moderately sized, with new features developed by the community regularly. However, the small number of permanent staff means that manual testing capabilities are limited. To facilitate regular and frequent integrations and deployments, automated testing plays an essential role in the Voluntarily software development lifecycle (SLDC).

Voluntarily's test suite is separated into three main categories; React component unit tests, Express API controller unit tests and API ability integration tests. The Ava[8] testing framework was used to execute these test suites. The entire test suite is run on GitHub Actions, a Continuous Integration managed service, before every integration.

React component unit tests used Enzyme[9] to mount the components in a testing environment, and generally assert that the visual elements are rendered correctly. The Sinon[10] test double library was used to mock out dependencies, increasing test isolation and controllability. The fetch-mock[11] library was used to verify that React components behave as expected when subjected to various server responses.

[8] https://github.com/avajs/ava
[9] https://github.com/enzymejs/enzyme
[10] https://github.com/sinonjs/sinon
[11] https://github.com/wheresrhys/fetch-mock

API controller unit tests verified that the controller methods behaved as expected, in isolation from the entire request pipeline. The mongodb-memory-server[12] library was used to provide a configurable, sandbox persistence environment for each test case. Document fixtures were defined and created for each test case. The mock-express-request[13] and mock-express-response[14] libraries were used to configure the request parameters and to assert the controller response.

API ability integration tests invoked the entire request pipeline and verified the actions that each user role was authorised for. Ability tests involved the creation of user fixtures with various roles and authenticating each request with a JSON Web Token (JWT) belonging to that user. For example, integration test cases were written to verify that an authenticated user could edit their submitted feedback, but not edit feedback submitted by others.

8. Methodologies

8.1. Project Management

The development of the data module was restricted by a fixed deadline (submission date) and fixed resources (the team). The Agile software development methodology supports fixed time and resource constraints in exchange for variable scope. Agile supported the project's constraints and was adopted by the team.

The Kanban framework for agile software development was used to organise development tasks and to help visualise the project's process. Jira's digital Kanban board allowed user stories to be created and visually tracked throughout their development lifecycle. Jira's integration with GitHub connects user stories with associated commits and pull requests.

Regular meetings were held with the Voluntarily team (primarily Andrew Watkins and Walter Lim), for feedback on developed features and updates to the project requirements.

[12] https://github.com/nodkz/mongodb-memory-server
[13] https://github.com/lykmapipo/mock-express-request
[14] https://github.com/lykmapipo/mock-express-response

8.2. Source Control Management

Source Control Management (SCM) is a significant factor in the success of medium-to-large sized projects with multiple developers. Poor SCM can result in difficult merge conflicts and frequently broken builds, both of which hinder development velocity. The shared repository model was used to facilitate safe parallel development while minimising source control overhead costs. This model involves collaboration from multiple developers to a shared repository, on separate branches.

The main development branch (master) was locked to ensure its consistency and quality. Pull Requests (PR) had to be submitted from a feature branch to modify the code on master. Submitted PRs had to be approved by at least one merger and successfully built and tested on the CI server.

8.3. Pair Programming

The majority of the data module implementation was completed following the pair programming practice. The author and his partner would take turns in screen sharing over Zoom, with one member developing and the other supervising.

Pair programming provided many benefits throughout the implementation process. Initially, the Voluntarily codebase was foreign and challenging to navigate. Collaboration from two developers flattened the learning curve of the codebase, allowing for faster development. With constant review from another programmer, coding mistakes are often identified much earlier, reducing the cost of fixing them. Lastly, pair programming proved to be an effective knowledge sharing strategy, where the author and his partner learnt good programming habits from each other.

9. Discussion

9.1. Reflection on Research Intent

The purpose of this research project was to construct a data module on the Voluntarily platform to measure the success of volunteering activities. This goal has been achieved through the development of the reporting dashboard. Using this feature, OAs can quickly view quantitative reports for volunteering activity. Reports include how many volunteers participated, how many hours were spent volunteering, where the volunteers participated and what activities they participated in.

In addition to the reporting dashboard, a feedback collection mechanism was developed to measure success from the volunteer's perspective. This mechanism provides an extensible platform for collecting quantitative and qualitative data from volunteers.

In addition to the functional project outcomes, the project was constrained by various non-functional requirements. Application performance was emphasised, to minimise hosting costs and to improve user experience. The performance requirement was especially significant when designing the database queries for reporting volunteering activity. Aesthetic consistency was essential to provide a consistent and appealing look and feel to the applications users. This was achieved through iterative design prototyping and the use of the Ant Design component library. Project maintainability was also significant, as the project is open-source and relies heavily on community contributions. Project maintainability was ensured through writing reusable, convention following, high-quality code, accompanied with comprehensive unit test suites.

9.2. Reflection on Technologies Used

Several reflections on the choice of technologies can be made after several development iterations using these technologies. The MERN software stack facilitates rapid development initially, due to its gentle learning curve and global use of JS. Also, interactions with the database are intuitive due to mongoose's developer-friendly API. However, several facets of the MERN and JS ecosystem proved to be a hindrance to the data module's development.

JS's weakly typed nature often caused confusion with the parameter types for existing functions. Often, time was spent deducing these types from existing usages of function. In a strongly typed language, such as TypeScript or Java, the parameter types could have been immediately made known from the typed function signature.

MongoDB's lack of a strict document schema resulted in inconsistent documents in the database and required many conditionals in the application layer to account for these inconsistencies. For example, a date field on the opportunity document was initially free text, allowing for strings such as "4h", "four hours" and "half a day". However, this was later converted to an ISO-8601 duration, such as "P1Y2M10DT2H30M". The lack of database migrations used by the Voluntarily platform required conditional logic to handle all present and

past formats of data stored in each field. These inconsistencies result in code redundancy and limited the modifiability and maintainability of the platform.

Lastly, the tendency of Voluntarily developers of adding many, poorly maintained NPM dependencies to a JavaScript project often caused frustration when library documentation was lacking or outdated. For example, the mongoose-crudify[15] library has a total of 18 commits and was last updated in May 2020. Mongoose-crudify is used heavily throughout the platform; nearly for every API controller.

10. Conclusion

The research and development work undertaken in this project has provided the foundation for a highly functional data reporting module. Many of the structures implemented have been designed with a focus on extensibility; facilitating future development on top of these foundations.

The reporting dashboard could be extended by adding additional reporting components or adding further controls to the existing components. For example, time-series functionality could be added, so Voluntarily OAs can understand how volunteering success has changed over time. Alternatively, the functionality could be implemented to allow end-users to customise their reporting dashboards, selecting which reporting components they are interested in.

The feedback collection process can easily be extended to accept further user input. The feedback submission page could feature optional fields, augmenting the rating created on page load. Questions from Hackman and Oldman's Job Diagnostic Survey could be added to this page. The results from these questions can be aggregated to form the MPS, used to quantitatively track the volunteer motivation.

11. Acknowledgements

The author would like to thank their research partner, Harrison Leach, and their project supervisor, Andrew Meads, for assistance and guidance throughout the research process. The author would also like to thank Andrew Watkins and Walter Lim from the Voluntarily team for their guidance and feedback throughout the project.

[15] https://github.com/ryo718/mongoose-crudify

12. References

[1] "Pam fergusson charitable trust," https://www.pamfergusson.org.nz/, (Accessed on 09/21/2020).

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

[3] "Csr case studies - datacom," https://www.datacom.co.nz/Case-Studies/Corporate-Social-Responsibility/City-Missions.aspx, (Accessed on 09/19/2020).

[4] "Community connect — xero blog," https://www.xero.com/blog/2019/05/community-connect-making-a-difference/, (Accessed on 09/19/2020).

[5] 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.

[6] L. Meijs and E. Ten Hoorn, "No "one best" volunteer management and organizing: Two fundamentally different approaches," *Challenges in volunteer management*, pp. 29–50, 2008.

[7] 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.

[8] M. Musick and J. Wilson, "Volunteers: A social profile. indiana," 2008.

[9] 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.

[10] 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.

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

[12] D. A. Dillman, "Tailored design of mail and other self-administered surveys," *New York, NY: Willey-Interscience*, 2000.

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

[14] 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.

[15] S. S. Ajibade and A. Adediran, "An overview of big data visualization techniques in data mining," *International Journal of Computer Science and Information Technology Research*, vol. 4, no. 3, pp. 105–113, 2016.

[16] J. 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

[17] "Design principle: Consistency. the most known and most fragile design... — by anton nikolov — ux collective," https://uxdesign.cc/design-principle-consistency-6b0cf7e7339f, (Accessed on 09/19/2020).

[18] "Aggregation — mongodb manual," https://docs.mongodb.com/manual/aggregation/, (Accessed on 09/19/2020).