ENGR 489 (ENGINEERING PROJECT) 2023

Cricket Data Visualisation App

Kevin Jose

Abstract — Cricket is a multifaceted sport that offers a great wealth of online data for potential analysis and discoveries. However, this data often remains inaccessible and overwhelming due to its tabular format and sheer volume. This project's aim was to address the above challenge by designing and implementing a software system that provides an elegant solution. I designed a system that effectively visualises cricket batting data and then implemented a web application (as proof of concept) to enable users to explore batting data in men's ODI cricket. The significance of this data visualisation application lies in its potential to enrich users' understanding of cricket and its rich history. By uncovering patterns and trends within the data, users, whether seasoned cricket enthusiasts or newcomers to the sport, can gain a deeper appreciation for cricket. In summary, this project successfully met its core objectives, offering a powerful tool to unlock the hidden insights within the world of cricket statistics.

I. INTRODUCTION

Cricket is the second most popular sport in the world. According to [2], it is estimated that globally, around 2.5 billion people follow cricket. There are three different formats in which cricket is played: test cricket, one day international cricket (ODI) and twenty20 (T20) cricket. Each format is unique and provides different datasets. In this project, I focused on the batting data within the one-day international (ODI) format for men's cricket. While the current data focuses on men's cricket, this system is versatile and can also be applied to women's cricket data. The purpose of this project was to implement a web application that enables users to visualise batting data in men's ODI cricket. This dataset encompasses the career statistics of every male player in the history of ODI cricket. The target users for this system are cricket fans who want to generate insightful visualisations of ODI batting statistics, to learn more about cricket and its rich history. Although the primary focus was on the ODI format, I designed the system to be adaptable for visualising data from other formats of cricket.

ODI cricket has been played all around the world for 52 years. Since the first ODI match in 1971, there have been more than 4500 matches played by 29 different teams. Of the 29 teams, there are 12 official countries that participate in ODI cricket every year. The 12 countries include the following: Afghanistan, Australia, Bangladesh, England, India, Ireland, New Zealand, Pakistan, South Africa, Sri Lanka, West Indies and Zimbabwe. Cricket, as a sport, is greatly impacted by environmental factors more so than other sports. Unlike indoor sports, the environment and ground conditions are not controlled in the sport of cricket. Cricket grounds can vary in size and pitches can vary in condition, according to the country in which the ground is located. Some countries tend to have dry pitches while other countries tend to have green pitches. These are some of the many environmental aspects which have an impact on the players' performance and ultimately contribute to the match results.

The official dataset in ODI cricket history contains over 2500 players and each player contains 13 attributes which summarise the player's batting career. The attributes are as follows: *name, career-span, matches-played, innings, not-outs, runs, highest-score, batting-average, balls-faced, strike-rate, centuries, half-centuries and below-fifties*. A row in the dataset encapsulates all these attributes, where each value represents the player's average performance over their career. Some of these attributes may be foreign to those who are unfamiliar with the sport therefore I have defined every attribute within the appendix. With such a large and complex dataset, we need a system which condenses this data and makes it more presentable so that the average user can explore, interact and visualise the cricket statistics. In this project, I will investigate what functionalities should be incorporated into an engaging, interactive system for understanding batting information in ODI cricket.

In response to the above problem, the aim of this project was to: design a system that effectively visualises cricket batting data and then implement a web application (as proof of concept) that enables users to visualise men's ODI cricket batting data. In this project, I designed a system, developed a proof-of-concept prototype, evaluated the prototype, implemented the complete system, and finally, I performed a user-study evaluation.

This tool enables analysis of cricket data in the form of data visualisations. Data visualisation is an effective way of presenting information to end-users because it makes it much easier for the human brain to understand and pull insights from [1]. The application is interactive and intuitive; it allows users to query cricket data (relating to ODI batting) and then generate visual charts for analysis. The impacts of this data visualisation app are that users can easily learn more about cricket and its rich history by discovering patterns and trends in the data. This gives fans a greater appreciation for the sport, in addition to educating those who are new to cricket and wish to learn more.

The following are nine **system requirements** that were established during the project's initial stages:

Non-functional requirements:

R1. **Web-based application:** the app must be able to run in any major web browser without plugins.

R2. **User-friendly experience:** the user interface should be straightforward and intuitive, catering to cricket fans who may not be frequent or advanced users.

Functional requirements:

R3. **Retrieve data from a cricket data source:** The system should have the capability to fetch data from an online cricket data source, such as ESPN Statsguru, and store this data in its database. Furthermore, the system should perform regular updates from the external data source to ensure that the database remains current with the latest statistics.

R4. **User selects a dataset:** The system will offer a range of datasets for the user's selection, allowing them to choose the dataset they wish to visualise. The selection of a dataset will be based on two key criteria: the cricket format and the gender associated with the data.

R5. **User creates a chart:** this feature allows the user to create charts by querying the dataset. By specifying parameters, the user can generate visual representations of cricket statistics.

R6. **User filters a chart:** this feature enables the user to apply filters to a chart, specifying criteria to narrow down the data displayed. Filtering enhances data exploration and the ability to focus on aspects of cricket statistics.

R7. **User compares charts:** this feature enables the user to compare multiple charts by viewing them simultaneously, enabling the analysis of various players or metrics side-by-side – for better insights.

R8. **User searches for a player:** this functionality allows the user to search for a specific player within the dataset, making it easier to access individual player statistics.

R9. **User views a player profile:** this functionality allows the user to access a player's profile, presenting detailed information about their ODI career performance.

User Personas

The target users of the system are cricket fans who wish to learn more about cricket and its rich history. I have created two user personas as examples of the target end-users of this system: Ricky Baker and Raj Singh.

Primary User

Name: Ricky Baker
Age: 19
Occupation: Tertiary Student at VUW
Bio: Ricky Baker is a second-year commerce student, studying at Victoria University of Wellington. Ricky is passionate about accounting and finance and he is working hard to pursue a career within this sector. He goes to the university campus 5 days a week to attend his lectures and study during the day. Cricket is another passion of his. Ricky is an active member of the university cricket team, in which he participates in training sessions and a cricket match every week. He is very close with his cricket teammates and he enjoys watching live cricket matches with them.
Expectations: When watching cricket matches, Ricky and his friends argue about their opinions on players and teams and who performs better. The internet provides a lot of tabular data on cricket statistics but Ricky is looking for a quick and interactive tool which provides simple visualisations of cricket data. During matches, Ricky wishes to instantly generate visualisations that will help him prove his point when making arguments to his friends about cricket. For example, he might use a visualisation that compares the batting performance of all players in ODI history in order to prove that Australian batsmen have generally performed better than New Zealand batsmen. Ricky will only use this tool occasionally and does not intend to spend large amounts of time on it therefore the tool must be simple and intuitive.

Secondary user

Name: Raj Singh
Age: 40
Occupation: PE teacher
Bio: Raj Singh is a high school PE teacher at St. Patrick's College in Silverstream. Raj is passionate about fitness, exercise and physical wellbeing. He works hard and takes his job very seriously because he believes in the importance of education and physical health. In his childhood days, Raj played a lot of cricket with his friends and he developed a passion for cricket. Although Raj does not play cricket anymore, he enjoys researching and following the sport and educating his students about it.
Expectations: Raj recently started a social media page with the goal of educating his students and wider community about cricket. On this page, he posts fun and interesting facts about history, news and statistics in cricket. The internet provides a lot of tabular data on cricket statistics but this data is uninteresting and not presentable to Raj's audience. He is looking for a quick and interactive tool which provides simple visualisations of cricket data. Raj realises the complexity of cricket data therefore he wants an intuitive system that will enable him to generate visualisations of all sorts of cricket statistics. Raj intends to use these visualisations in his social-media posts in order to make his content more interesting, informative and insightful. Raj will use this tool to gather information for his content and he does not intend to spend large amounts on it therefore the tool must be simple and intuitive.

II. Related Work

This section discusses the related work and background research of the project. We will explore some existing solutions of sports data visualisation tools, making comparisons with this system. In addition, we will discuss some basic visualisation theory, which is crucial in determining how the system is designed.

A. Statsguru

Cricket is a complex sport which provides a great wealth of data (through ESPN cricinfo) but this data is not very accessible or digestible in its given format. Statsguru [3] (by ESPN cricinfo) is an online tool which allows a user to query data from its cricket statistics database. After the user submits a query, Statsguru returns a table of data which contains all the resulting rows. Statsguru contains the largest online database of officially recorded data in ODI cricket. Its data ranges from the beginning of ODI cricket (1971) to the present. This tool is powerful in the sense that it allows users to choose from a range of filters and it contains all of the data in the history of ODI cricket.

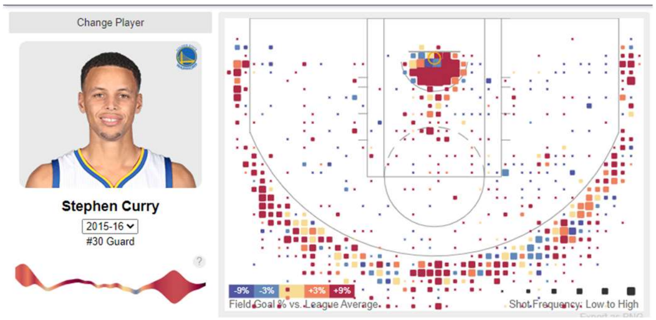
Although Statsguru is a good data querying tool, its limitation is that it does not provide any features for data visualisation and therefore limits the potential for data analysis and discoveries. Statsguru returns a table of raw data and this can be difficult to analyse and comprehend because of the complexity and factors involved in the sport. Due to the lack of data visualisation (such as graphs) in this tool, users are missing out on a lot of learning and findings from this data. In the last 52 years of ODI cricket, there have been many factors that have changed and evolved the sport [4]. Some of these factors include: rule changes, improvements in equipment, improved sensor technology. In order to discover these factors and their effects on player and team batting performance, users must be able to visualise the data in the form of charts.

B. Example Systems

The following are existing examples of sports data visualisation tools that I will discuss. I will give a brief overview of each system and then discuss the differences between the example system and my system.

Buckets

Buckets is an online data visualisation tool for NBA basketball players [6]. Buckets allows users to analyse the statistics of individual NBA players. Initially, the user must select a player they would like to analyse then they must select the NBA season. Once the above selections are made, the system displays three main visualisations: a 'court map' chart, a 'shot-frequency percentage' chart, and a 'field-goal percentage' chart.



The following are differences between the Buckets system and my system. Firstly, the Buckets system analyses player statistics for a specified season of NBA basketball because its purpose is to analyse at a closer level. Comparatively, my system visualises a player's performance at a higher level by using the player's average statistics achieved over their career. The purpose of my system is to analyse player and team performance at a historical level whereas the Buckets tool analyses player performance in each specific season.

Another significant difference is that Buckets uses player positioning in all three of its visualisations. All of the visualisations display the distance (from the hoop) at which the player attempted to shoot the ball and the frequency of each distance. The shot-frequency % chart shows how frequently a player attempts a shot compared to his teammates, at different distances. The field-goal % chart shows the player's ratio of field goals scored to field goals attempted, at different distances. The court map chart shows the court and the precise positions at which shot attempts were made, during the season. It visualises a combination of the information in the two former charts. Comparatively, my system does not visualise any information about a player's positioning on the cricket ground because this kind of data is not relevant for analysing a batsman's performance in cricket.

One point of inspiration that I recreated in my system is the player-profile panel from Buckets. In my system, a data point represents a cricket player. Therefore, if the user hovers on a data point, the corresponding cricket player's details are displayed in the profile panel. In addition, a player's details can be viewed by using the player-search functionality. The player-profile panel turned out to be a crucial feature in the system, creating a more engaging and interactive user experience.

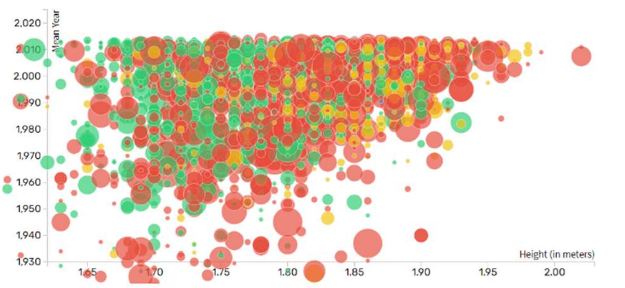
Cricket Visualiser by Lascarides

Cricket visualiser is a basic visualisation tool created by a cricket fan named Michael Lascarides [5]. The tool visualises cricket matches that took place in the 2015 ICC World Cup. Lascarides created this tool as a personal IT project.

The following are differences between the example system and my system. Firstly, the example system focuses on visualising matches that occurred during the 2015 World Cup tournament whereas my system focuses on visualising player and team batting performance in ODI cricket history. Due to the different focuses in these systems, they visualise very different types of data. The example system is very specific because it visualises every play that occurs in a cricket match. Using Manhattan graphs, it shows every event that occurs ball by ball, for every over in the match. Contrastingly, my system does not visualise specific events that occurred in each cricket match because the system works with high-level data in order to give an overview of players' careers and team performance. For example, my system includes scatter-chart visualisations in order to visualise the relationships between players and compare their career performance.

Football Visualiser by Dhvanil

Football visualiser is another visualisation tool created by a football fan named Dhvanil [7]. The nature of the data and visualisations in this tool is similar to those in my system. Dhvanil created the visualisation tool as a personal project.



As shown in above image, the example system visualises all 20000 football players (from its dataset) within a scatterplot visualisation. This visualisation is very important for the example system because it shows all of the data involved in the system and also provides meaningful filtering options in order to narrow down the information. This scatterplot visualisation is very comparable to the scatterplot visualisation within my system prototype. Both visualisations share the purpose of comparing player performance for international careers in the history of the sport. In the football visualiser, the dataset dates back to the first football world cup in 1930. In the cricket visualiser (my system), the dataset dates back to the first ODI cricket match in 1971. The scatterplot within the football visualiser is very insightful because it contains a lot of information within the single chart. In addition, the filtering options are easy and functional for narrowing down the information in order to find all sorts of patterns and trends.

Comparatively, the scatterplot chart within my system prototype is much simpler and less informative than the one in the example system. The following are a few aspects in the football scatterplot that I initially intended to take as inspiration and recreate in my system's scatterplot. Firstly, the football scatterplot uses size variation of data points in order to compare performance of different players. For instance, players who have scored more goals will be represented by larger data points on the chart. Secondly, the example scatterplot provides filtering options by which the user can select what attribute will affect the size of the data points. For instance, if the user selects the 'Appearances' attribute, then players who have had more international appearances will appear as larger data points. Thirdly, the football scatterplot includes a country filter which allows the user to visualise players who are from a specific country. The above are three points of inspiration that I intended to implement in my system but I could only recreate one due to time constraints. I recreated the third feature: filtering data by a country.

C. Visualisation Theory

The Eyes Have it

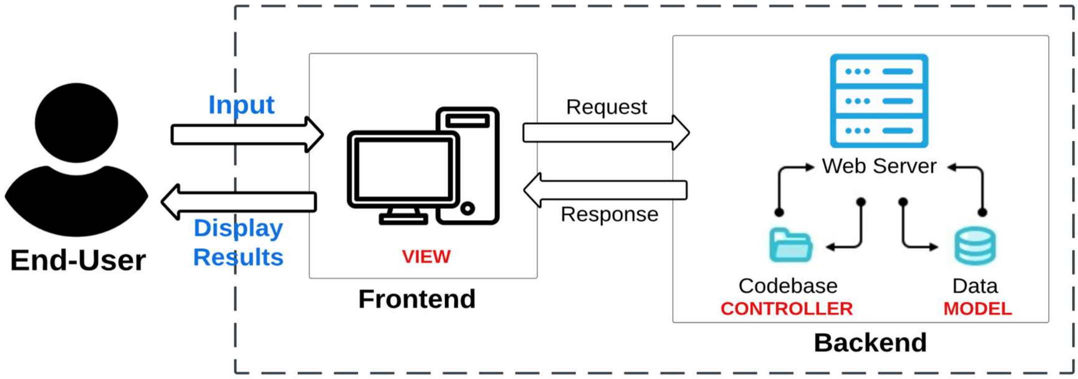
'The Eyes Have it' by Ben Shneiderman [8] is a fundamental text that I studied during the background research phase of the project. This text provided important background information which equipped me with necessary knowledge for planning and designing effective visualisations. This text is an essential guide that teaches how to design and create effective, engaging and purposeful visualisations within graphical user interfaces.

In the text, Shneiderman introduces and explains his taxonomy for information visualisations based on the types of data being presented and the tasks that users want to achieve. The goal is to provide a framework that helps designers and researchers to create effective visualisations. Shneiderman points out that understanding and classifying the data type (in the system) is crucial for choosing the appropriate visualisation techniques. The text also discusses the tasks that users commonly perform when interacting with visualisations. The tasks are the following: searching, querying, comparing, organising, and analysing data. Shneiderman recognises that the alignment of relevant tasks and data types enables designers to create visualisations that are more intuitive and effective towards the users' goals.

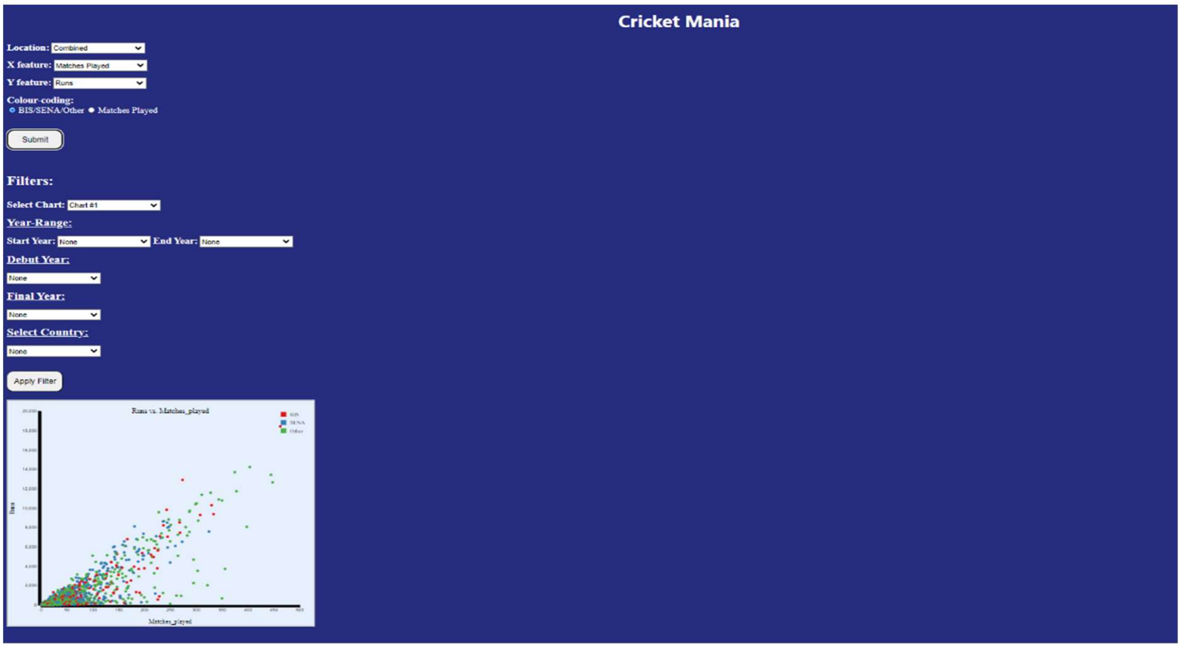
Shneiderman proposes a layered approach by which visualisations should be presented to the end-users. This layered approach includes the following: overview first, zoom and filter, and details on demand. Essentially, this concept suggests that users should be provided with a high-level overview of the data first, allowing them to zoom in, apply filters, and request specific details as needed. In my system, I implemented all of the above, except for a zooming feature. Zooming is a feature that I would highly consider for future development.

In summary, the goal of Shneiderman's taxonomy is to provide a framework for understanding data types in order to design effective visualisations. By considering the tasks users want to accomplish and aligning them with appropriate visual representation, designers can create visualisations that improve data exploration and analysis.

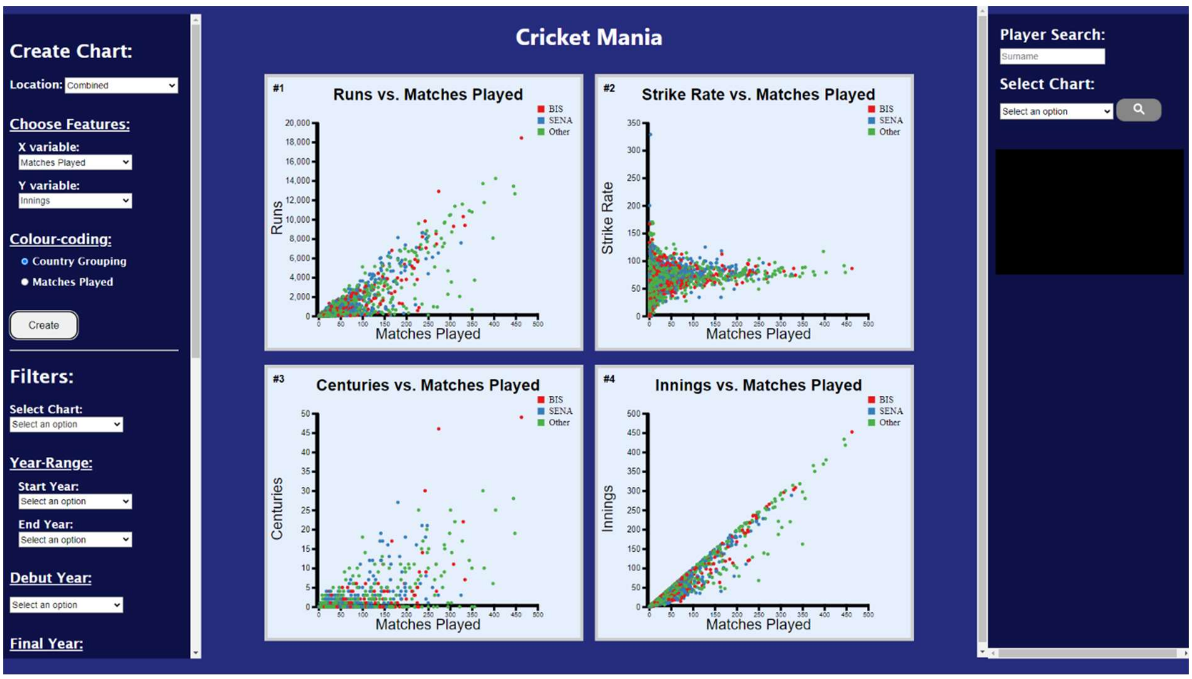
System MVC Architecture



UI Design #1



UI Design #3



III. DESIGN

This section reveals the design of the cricket visualisation system, in terms of **system architecture**, **user interface**, and **sustainability considerations**. To recap, the system's functional requirements encompass a set of key operations: importing data, generating charts, applying chart filters, conducting chart comparisons, and a player search functionality. These specific requirements played a pivotal role in shaping both the system's architecture and its user interface design.

A. System Architecture

I designed this system according to the MVC design pattern. The MVC architecture is a common design pattern that separates an application into three interconnected components: model, view, and controller.

In the previous page, there is a UML diagram that displays the high-level architecture design of the system. I have indicated the following MVC components in the diagram: the 'view' is the user-interface, The 'model' is the data, and the 'controller' is the app logic – that is within the codebase.

We will now explore the general steps involved in creating a chart within this system. The user initiates the process by interacting with the user interface (view), where they input the necessary parameter values for the chart. Once the user has provided these inputs, they submit the data, which is then bundled and transmitted to the web server. Upon receiving this request, the web server takes charge of handling it and directs it to the controller. Within the controller, the values from the request are unpacked, and data is queried from the dataset. The controller then retrieves the required data, compiles it into a structured response, and sends it back to the user interface via the web server. Finally, the resulting chart is displayed to the user (on screen), for analysis.

This illustration serves to elucidate the intricate relationships between the various components of the MVC architecture that underpins this system.

B. User Interface Design

The user interface (UI) of the system evolved over the course of three significant iterations. As I developed new functionalities, I discovered various design challenges that required the UI to be redesigned. Note that I have only included screenshots for Design #1 and Design #3, due to limited space.

Design #1 (image in prev. page)

In the initial design iteration, the page elements follow a simplistic linear arrangement. The form is presented as a straightforward list of inputs, and charts are added at the bottom of the page as they are generated. This arrangement necessitates extensive vertical scrolling to navigate the system. Users have to scroll down to view their generated graphs and then scroll up to apply filters or generate a new chart. This webpage structure results in a poor user experience due to lack of navigability, limited visibility, and the need for excessive scrolling.

Design #2

In the intermediate design iteration, I undertook a significant restructuring of the webpage by dividing it into three distinct vertical regions. This update aimed to address the navigation and visibility issues present in the initial design. The layout was thoughtfully crafted so that each region had a specific purpose, ensuring a clear and easily navigable structure for users. However, it's important to note that this iteration came with a limitation: users could only view one chart at a time, which restricted their ability to make direct chart comparisons.

Design #3 (image in prev. page)

In the final design iteration, I tackled the aforementioned limitation by optimising chart size and adopting a 4x4 grid layout for chart presentation. This strategic modification played a pivotal role in meeting the functional requirement R7 – comparing charts. In this redesigned interface, users can simultaneously view up to four charts - offering enhanced comparability options for in-depth analysis and the discovery of valuable insights.

C. Sustainability Considerations

While this project has a specific emphasis on batting data in men's ODI cricket, I have designed a versatile system with the capability to visualise data from women's cricket and other formats of the game, including T20 and test matches. The system incorporates a comprehensive set of data attributes, including *name, career-span, matches played, innings, not-outs, runs, highest score, batting average, balls faced, strike rate, centuries, half-centuries, and below-fifties*. Significantly, these 13 attributes are universally applicable to other formats of cricket, enabling the system to visualise datasets beyond just men's ODI cricket.

The system's adaptability, coupled with its data-importing functionality, contributes to its sustainability from a social perspective. Its design empowers users to import cricket batting data of their choosing, including datasets related to women's cricket. Consequently, users can analyse and visualise statistics for female cricket players, making the system socially impactful by promoting greater gender inclusivity in the sport of cricket. This capacity to shed light on women's participation in sports is instrumental in raising awareness about and advocating for increased female representation in sports. This aligns with the UN sustainability development goal #5: to achieve gender equality and empower all women and girls [9]. Given the historical underrepresentation of women in sports, it is essential to raise awareness and encourage their active involvement in the sporting world.

From an environmental standpoint, this system raises no concerns since it is not a computationally demanding application. The system's functionalities are neither overly complex nor power-intensive, which mitigates worries about excessive energy consumption. Data retrieval occurs infrequently, and database queries are straightforward. Furthermore, the built-in data is minimal in size, typically in the range of kilobytes, ensuring highly efficient data processing and management. This aligns with the UN sustainability development goal #12: to ensure sustainable consumption and production patterns [10].

Looking into the future, it's important to consider the potential for widespread adoption of this system if it were to be commercialized or gained viral popularity, particularly given the substantial global following of cricket as a sport. In the event of the application going viral, it could attract millions of potential users. At such a mass-scale usage, there arises a significant concern related to increased energy consumption and its associated carbon emissions.

IV. Implementation

This section delves into the system's implementation, providing an insight into the approaches taken to develop the application's major components. It encompasses the implementation of the application in the following key areas: data preparation, essential technologies, and the engineering challenges faced.

A. Data Preparation

Data Gathering

Data preparation was a significant phase in the development of this project. Firstly, I used Statsguru to query the necessary data for the system. All the data that I gathered from Statsguru is publicly available information. I made a total of 6 queries for player statistics filtered by the following locations: home, away, neutral, total, BIS[1] and SENA[2] . Filtering by location gives a player's statistics in the specified location. For instance, if the specified location is 'home', then all resulting data rows would give a player's statistics, based on matches played in home grounds. I extracted the resulting tabular data from Statsguru and pasted this into an Excel spreadsheet. After gathering data from the 6 queries, the Excel dataset contained statistics for around 2800 players and each player contained up to 6 rows of data. Then I exported the spreadsheet as a CSV file so that it could be ready for data processing.

Data Processing

The data processing phase involved inputting the raw data from the CSV file and outputting JavaScript objects. Firstly, I used D3's csv() function to read the CSV file and return an array of objects. There were around 13000 data objects extracted from the CSV file and an object represented a row of data from the original dataset. This data was processed so that objects were grouped by the player which they belonged to. For instance, if the system read 6 objects containing the name 'A Bagai', then these objects would get grouped together under a parent object of the identifier 'A Bagai'. After the processing was complete, the web app stored this data in an array of player objects.

B. D3 JavaScript Library

After researching options and discussing with my supervisor, I decided to use the D3.js technology in my system. D3 is a popular JavaScript library used for creating dynamic and interactive data visualisations on the web. It provides a powerful set of tools for manipulating documents based on data, allowing developers to create visually appealing and interactive representations of the data.

D3 was effective for this system because it had:

- **A data-driven approach:** D3 was purpose-built for binding data to a Document Object Model (DOM) and applying data-driven transformations. This allowed me to seamlessly map data with webpage elements, resulting in dynamic updates as the data changed. For instance, when a user applies a chart filter, the chart dynamically adjusts its data points to display the filtered data, creating a responsive and intuitive interface, thus meeting requirement R2.
- **Customisation:** D3 provided fine-grained control over every aspect of visualisations. It enabled me to create custom charts and graphs that tailored to this application's unique requirements.
- **Powerful data transformation:** the library offered a wide array of tools for data manipulation, which proved invaluable for transforming and preparing the data for visualisations. Using the provided tools, I was able to effortlessly perform operations like filtering, grouping, and sorting on the data, among many other options.
- **Transitions and animations:** it offered built-in support for smooth transitions and engaging animations. These features enhanced the interactivity of visualisations, empowering end-users to explore and experiment with data, ultimately facilitating insightful discoveries.
- **A vast community and ecosystem:** D3 boasts a vibrant and active community, offering a plethora of tutorials, documentation, forums, and online resources. This rich array of support and inspiration opportunities was greatly beneficial as I was utilising the library.

[1] Bangladesh, India, Sri Lanka
[2] South Africa, England, NZ, Australia

C. Vanilla JavaScript

This application was built using Vanilla JavaScript (JS) technologies, namely HTML, CSS, and JavaScript, with D3 being the only library involved. I chose to use this combination of Vanilla JS and D3 because it was the approach that suited to the focus of the system. This system's primary focus is to produce interactive visualisations to enable users to explore cricket statistics and discover insights. As established in the non-functional requirements R1 and R2, this system must be web-based and provide a user-friendly experience. The selected combination of technologies fulfils these requirements while staying true to the system's primary focus.

There are alternative technologies that I considered such as the React framework and Fluent UI but after careful comparing and contrasting, I made a final decision to go with vanilla JS and D3.

D. Engineering Challenges

The following are some significant engineering challenges that I experienced in the development of the system: **lack of software modularity** and **management of user input data**.

Lack of Software Modularity

One of the significant challenges I encountered was the lack of software modularity. Initially, I wrote extensive code in the early development stages, which lacked modularity and had oversized functions. This approach allowed for faster initial code creation. However, when I revisited this code during later stages of development, I faced significant difficulties. Comprehending the code, fixing bugs, and implementing new functionalities became daunting tasks due to the absence of a concise and modular structure. In the long run, this approach resulted in time wastage as a substantial amount of resources was spent on building new features and resolving simple bugs. These tasks took far more time than anticipated, primarily because of the dense and monolithic nature of the code. Consequently, this led to unplanned work, necessitating extensive refactoring and restructuring to enhance code readability, maintainability, and reusability.

To address these challenges in the future, I would make the following changes to my approach. Firstly, I would enhance my software architecture planning by creating a use-case diagram to proactively define classes and essential functions of the code. It would be advantageous to allocate more time during the project's early stages for in-depth architecture planning at both high and low levels. While I only focused on high-level architecture planning in this project, I believe that incorporating low-level planning is also important. Additionally, I would establish a practical set of coding standards to ensure the creation of efficient and clean code. To further bolster code quality, I would consider using a static analysis tool, which would help verify that the code aligns with industry standards.

Management of User Input Data

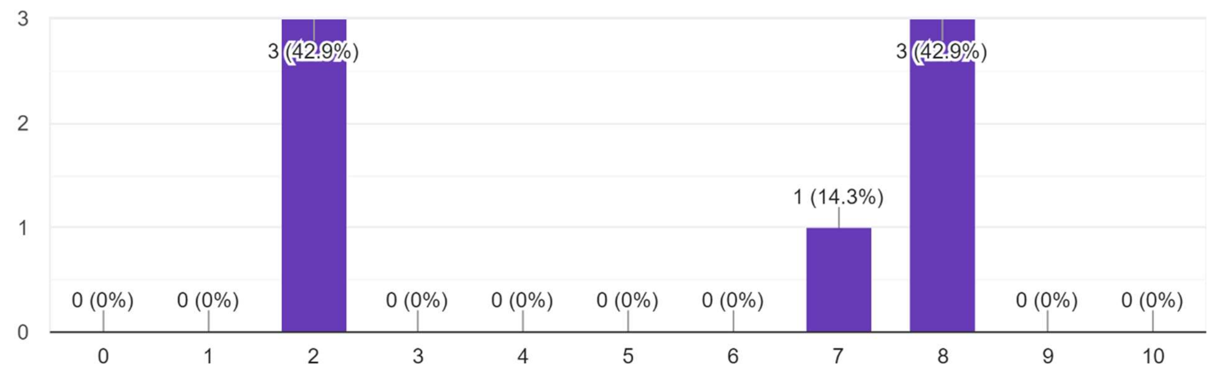
Another challenge I encountered revolved around managing and processing user input data within the application. To address this, I made a pivotal change to how I processed user input data by incorporating HTML form elements. The introduction of the form element was a critical turning point. Initially, I didn't use forms; instead, I had separate elements for each input, and I had to set up event listeners for each one. This initial approach proved to be inefficient, time-consuming, and challenging to manage user inputs effectively. After researching various approaches for gathering user input, I decided to adopt the method that leveraged the HTML form element.

The form element is designed to help developers structure, organize, and process user input information within a webpage seamlessly. In a JavaScript function responsible for form processing, I used querySelector to access the form and converted it into a FormData object. This FormData object conveniently constructs a set of key/value pairs that represent form fields and their values. FormData proved to be exceptionally useful for form processing and chart configuration, allowing me to extract the necessary chart specifications and generate charts accordingly.

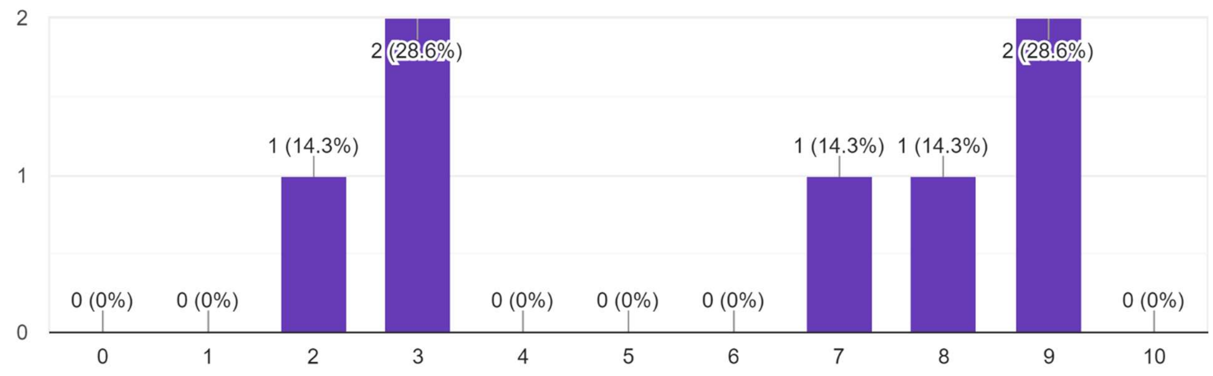
In the future, I will ensure to conduct thorough research before embarking on the development of a major application component. This proactive approach would have saved a significant amount of time, as it would have familiarized me with the diverse methods of form processing from the outset.

Pre-Test Questionnaire Results

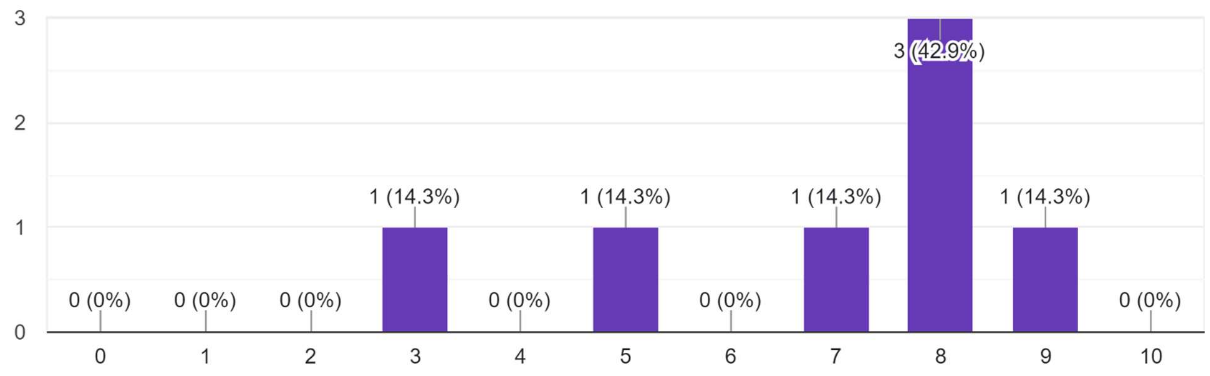
What is your level of cricket knowledge? (ie. rules, history, players, etc.)
7 responses



What is your level of interest in cricket?
7 responses



How interested are you in sports?
7 responses



V. EVALUATION

In the final stages of the project, I conducted user study testing to evaluate the implemented system. The purpose of this evaluation was to:

- Assess the system's usability – the quality of the user experience
- Validate whether the software meets real user needs and expectations
- Identify common errors and obstacles
- Collect user feedback – opinions, suggestions, criticisms

A. Testing Participants

The evaluation involved seven user-testing participants. The demographic of this group comprised fourth-year engineering students, all of whom were male and aged between 21 and 23.

The pre-test questionnaire results (on next page) depicted a notable distribution among the participants. Specifically, 57% of respondents exhibited a high level of confidence in their cricket knowledge, while the remaining 43% expressed limited knowledge. Furthermore, a similar split emerged in terms of interest, with 57% displaying a strong affinity for cricket and the remaining 43% indicating lower interest. Taking a broader perspective on measuring sports-related interest, it's noteworthy that 71% of the participants showcased strong enthusiasm for sports, while the remaining 29% were positioned in the category of low-to-moderate interest. These findings were instrumental in providing a clear understanding of the participants' backgrounds, particularly their association with cricket and sports in general.

The visual representations in the charts underscore the substantial diversity in participants' levels of cricket knowledge and interest. This diversity was essential in ensuring a more robust and equitable evaluation of the system. It allowed for feedback and results to be collected from users with varying degrees of knowledge, enabling a more comprehensive perspective on the system's performance.

B. User Testing Process

I will now outline the process that I followed to conduct user testing on each participant. The user testing was broken into three stages: pre-test questionnaire, system trial, and post-test questionnaire. I conducted a total of seven testing sessions for the seven participants. Each session ran for a duration of up to 45 minutes. The testing was done on a Linux PC in lab room CO228.

Pre-Test Questionnaire

The session started with participants filling out a pre-test questionnaire using Google Forms. This survey aimed to assess their backgrounds, specifically their association in cricket and sports in general. This survey consisted of three Likert scale questions to gauge the:

- User's level of knowledge in cricket
- User's level of interest in cricket
- User's level of interest in sports

System Trial

The most significant part of the session was trialling the system. It commenced with a user briefing, where I introduced the application's purpose and provided contextual information about the data involved. To ensure comprehensive feedback, I equipped the user with a notepad for comments and suggestions. Additionally, I had a document ready to record the user's responses to verbal questions.

The trial comprised two rounds. The first round offered a straightforward, introductory user experience, aimed at acquainting the user with the system. Detailed, specific instructions guided the user through the creation of a basic chart and the use of two key functionalities for exploring data. I posed three simple questions to confirm the user's understanding of the system's fundamental features.

The second round presented a more complex challenge, involving two in-depth user journeys for detailed data analysis. The instructions were less prescriptive, requiring the user to rely on the interface design and heuristics to perform tasks. The first journey involved an investigation of player strike-rate performance over decades. The second journey examined century-scoring performance of players from four different countries. During these journeys, participants were asked verbal questions, necessitating analysis and judgment. It was fascinating to observe the diverse approaches taken by participants. This revealed areas in the system that led to differing interpretations and approaches, shedding light on functionalities that might need improvement for better user comprehension.

Post-Test Questionnaire

Following the completion of the system trial, I administered a final questionnaire to the participants, asking them to reflect on their experience using the system and provide comments and feedback. This questionnaire was created using Google Forms and consisted of 11 Likert scale questions. Additionally, there were three open-ended questions for participants to share more detailed feedback. The Likert scale questions were modelled after the System Usability Scale (SUS) questionnaire, a widely recognized tool for assessing software system usability. I opted for the SUS questionnaire due to its suitability for small sample sizes, its clear and direct questions, and its ability to provide a quick assessment of the system's usability.

C. *User Testing Results*

The user testing results are revealed on the next page, which are the results from the post-test questionnaire.

SUS Results

The results from the Likert scale questions have unveiled several significant findings (note, questions are referred to in abbreviated form, eg. *Q5 for question 5*):

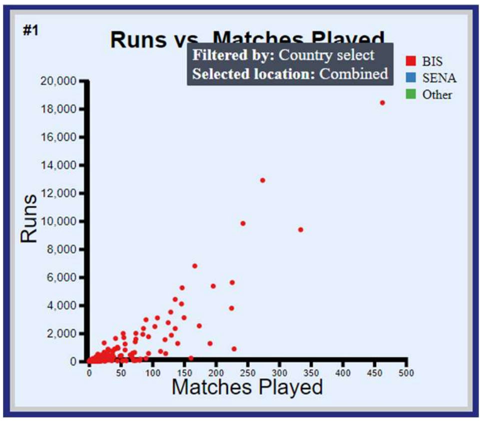
- Firstly, the Q2 chart indicates that 86% of the participants found the system to be simple and straightforward. This achievement aligns with system requirement R2 - a user-friendly interface.
- Secondly, the Q3 results demonstrate that 71% of users found the system easy to use for visualising and analysing cricket statistics, further reinforcing the fulfilment of R2.
- Thirdly, Q9 reveals that 86% of users expressed a high level of confidence in using the system. Only one user provided a neutral response, indicating a generally positive sentiment.
- Finally, Q11 indicates that 100% of the participants believed that this system is a viable tool for visualising and analysing historical cricket data. This outcome signifies the accomplishment of requirements R5-R9, all of which are crucial for a viable cricket data visualisation system.

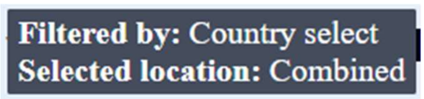
Feedback on System Usability

The following are common points of feedback from the testing participants for improving the user experience of the system.

Problem #1:

First and foremost, the most prevalent issue identified was the absence of filter-query values being displayed after applying a filter to a chart. While the charts indicated the type of filters applied, they failed to present the specific values associated with those filters.





The images above illustrate this problem. In the chart's pop-up, additional data details are provided to the user. In this case, the chart displays data that has been filtered by the 'Country select' filter. However, the actual names of the countries within the filtered data are not visible. This represents a UI design flaw that I had initially overlooked but was brought to my attention by keen-eyed participants. Remarkably, four participants pointed out this issue as a UI problem.

Problem #2:

Another issue of significance was the inability to delete or clear charts and the lack of an option to undo chart transformations, such as filtering. Consequently, when a user desired to remove a chart or reverse an action, the sole recourse available was to reload the entire application, which proved to be impractical. This issue was prominently highlighted by three participants as a common concern.

VI. Conclusions and Future Work

Future Work

In response to the feedback given in the evaluation, I would address the two problems by listing them as minor tasks in the system's future work. The first task would be to solve problem #1 by expanding the chart pop-up to indicate the filter-query values. The second task would be to solve problem #2 by incorporating two new functionalities to the system: deleting a chart and undoing an action on a chart.

There is one major task that I would like to see achieved in the next iteration of this project. This task is to achieve system requirement R3: to set up a dedicated system database and link it to an external cricket data source to regularly fetch data and keep the system's model updated with the latest cricket statistics. This functionality was included in my system design but due to time constraints, I decided not to implement it and chose an alternative data-retrieving approach.

Conclusions

The problem addressed by this project was: the cricket data (available online) often remains inaccessible and overwhelming due to its tabular format and sheer volume. The official dataset in ODI cricket history contains over 2500 players and each player contains 13 attributes which summarise the player's batting career. With such a large and complex dataset, a system is needed that condenses this data and makes it more presentable so that the average user can explore, interact and visualise the cricket statistics.

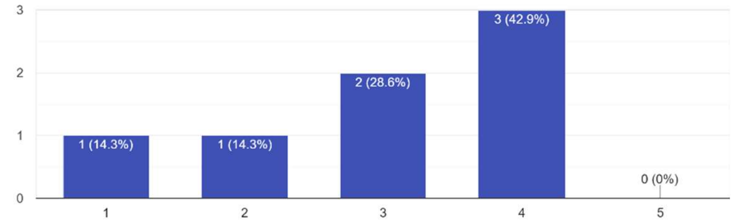
The goal of this project was to: design a system that effectively visualises cricket batting data and then implement a web application (as proof of concept) that enables users to visualise men's ODI cricket batting data.

In conclusion, this project was successful in achieving the goal. As evident in this report, the software design was successfully created for a complete system solution and the data visualisation application was successfully implemented.

Post-Test Questionnaire Results

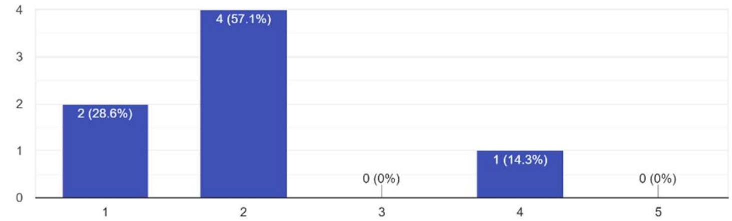
Q1: I think that I would like to use this system frequently

7 responses



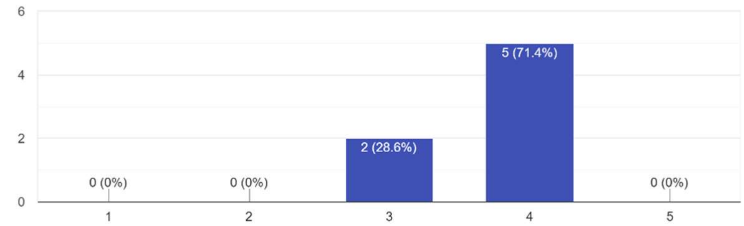
Q2: I found the system unnecessarily complex

7 responses



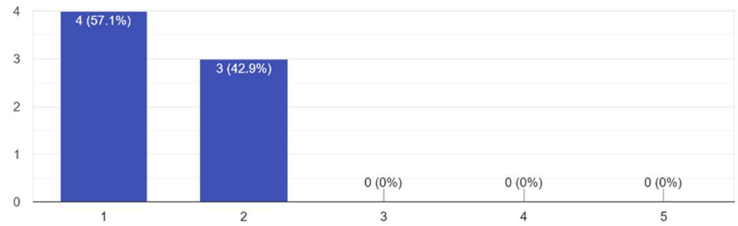
Q3: I thought the system was easy to use

7 responses



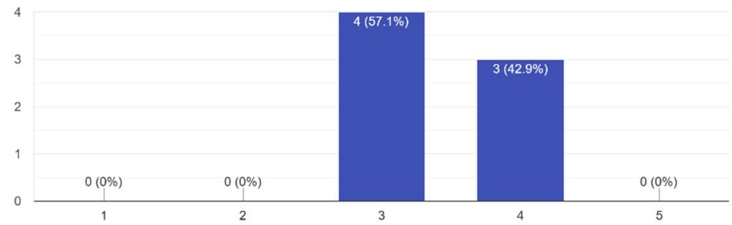
Q4: I think that I would need the support of a technical person to be able to use this system

7 responses



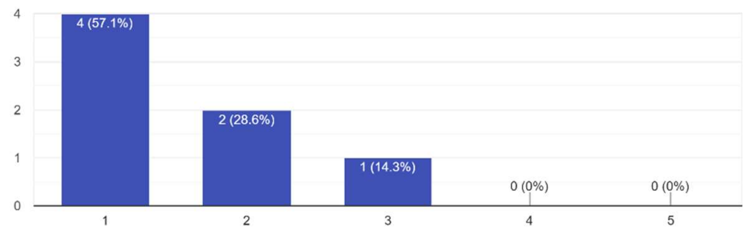
Q5: I found the various functions in this system were well integrated

7 responses



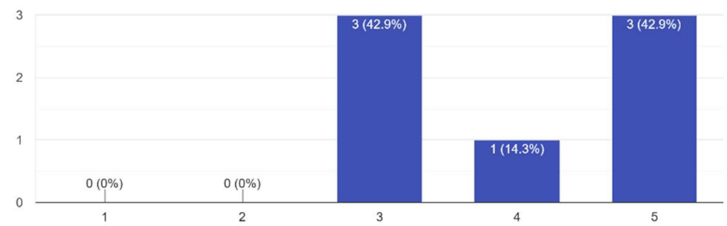
Q6: I thought there was too much inconsistency in this system

7 responses



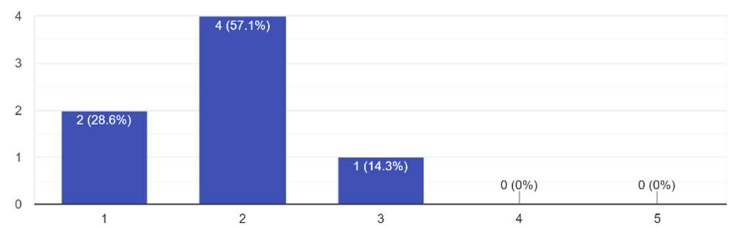
Q7: I would imagine that most people would learn to use this system very quickly

7 responses



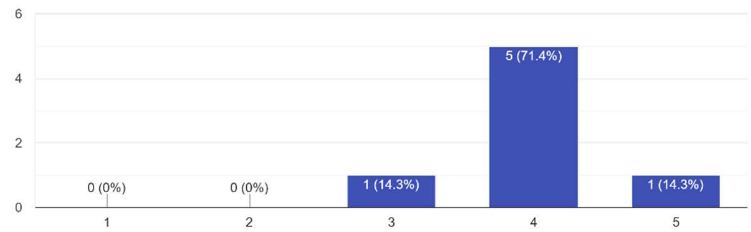
Q8: I found the system very complicated and inefficient to use

7 responses



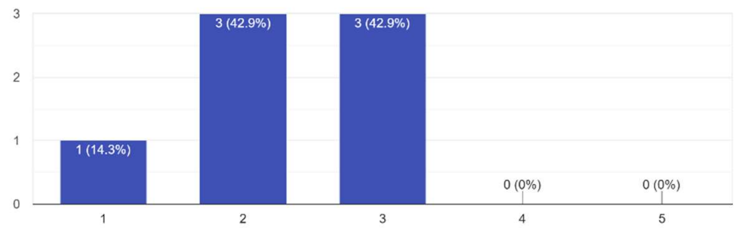
Q9: I felt very confident using the system

7 responses



Q10: I needed to learn a lot of things before I could get going with this system

7 responses



References

[1] Brush, K. and Burns, E. (2022) What is data visualization and why is it important?, Business Analytics. TechTarget. Available at: https://www.techtarget.com/searchbusinessanalytics/definition/data-visualization#:~:text=Data%20visualization%20is%20the%20practice,outliers%20in%20large%20data%20sets (Accessed: March 31, 2023).

[2] Smith, T., Sakr, M. and Worthington, J. (2022) How popular is cricket around the world? [statistics], SQaF. Available at: https://sqaf.club/how-popular-is-cricket/#:~:text=Is%20Cricket%20a%20Popular%20Sport,Pakistan%2C%20Sri%20Lanka%20and%20Bangladesh (Accessed: March 31, 2023).

[3] ESPN CricInfo: Statsguru (no date) Cricinfo. Available at: https://stats.espncricinfo.com/ci/engine/stats/index.html (Accessed: March 31, 2023).

[4] The evolution of ODI cricket | Andy Bull (2015) The Guardian. Guardian News and Media. Available at: https://www.theguardian.com/sport/2015/feb/10/the-evolution-of-odi-cricket (Accessed: March 31, 2023).

[5] Lascarides, Michael . “Visualizing Cricket | Michael Lascarides.” Lascarid.es, lascarid.es/cricket/index.html. Accessed 2 June 2023.

[6] Beshai, Peter. “Buckets: NBA Shot Visualization.” Buckets.peterbeshai.com, buckets.peterbeshai.com/app/#/playerView/201939_2015?countMin=1. Accessed 2 June 2023.

[7] Dhvanil. “Football Data Visualization.” Dhvanil.com, dhvanil.com/football.

[8] Shneiderman, B. “The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations.” Proceedings 1996 IEEE Symposium on Visual Languages, 2020, ieeexplore.ieee.org/document/545307/, https://doi.org/10.1109/vl.1996.545307.

[9] United Nations, “Goal 5,” sdgs.un.org, 2023. https://sdgs.un.org/goals/goal5

[10] “Goal 12 | Ensure Sustainable Consumption and Production Patterns.” United Nations, 2023, sdgs.un.org/goals/goal12.

Appendix

Appendix is included as a separate document in submission.

Kevin Jose (Student, VUW)
4th Year in Software Engineering