A

Project Report

on

Admin Dashboard Analysis:

A Web-based Statistical Dashboard

Submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology

by

M. Uday Kiran

(20EG105733)

S. Santosh

(20EG105560)

D. Reshwanth

(20EG105516)

Under the guidance of

Mrs. S. Bhagya Rekha

Assistant Professor,

Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANURAG UNIVERSITY

VENTAKAPUR (V), GHATKESAR (M), MEDCHAL (D), T.S - 500088

TELANGANA

(2023-2024)

DECLARATION

We hereby declare that the report entitled **"Admin Dashboard Analysis"** submitted to the **Anurag University** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology (B. Tech)** in **Computer Science and Engineering** is a record of an original work done by us under the guidance of **Mrs. S. Bhagya Rekha, Assistant Professor** and this report has not been submitted to any other university for the award of any other degree or diploma.

Place: Anurag University, Hyderabad

M .Uday Kiran
(20EG105733)
S. Santosh
(20EG105560)
D. Reshwanth
(20EG105516)

CERTIFICATE

This is to certify that the project report entitled **"Admin Dashboard Analysis"** being submitted by **M. Uday Kiran, S. Santosh, D. Reshwanth** bearing the Hall Ticket numbers **20EG105733**, **20EG105560, 20EG105516** respectively in partial fulfillment of the requirements for the award of the degree of the **Bachelor of Technology** in **Computer Science and Engineering** to **Anurag University** is a record of bonafide work carried out by them under my guidance and supervision from 2023 to 2024.

The results presented in this report have been verified and found to be satisfactory. The results embodied in this report have not been submitted to any other University for the award of any other degree or diploma.

Signature of The Supervisor
Mrs. S. Bhagya Rekha
Assistant Professor

Signature Dean CSE,
Dr. G. Vishnu Murthy
Department of CSE

External Examiner

ACKNOWLEDGMENT

We would like to express our sincere thanks and deep sense of gratitude to project supervisor **Mrs. S. Bhagya Rekha,** Assistant Professor, Department of Computer Science and Engineering, Anurag University for her constant encouragement and inspiring guidance without which this project could not have been completed. Her critical reviews and constructive comments improved my grasp of the subject and steered to the fruitful completion of the work. Her patience, guidance and encouragement made this project possible.

We would like to acknowledge our sincere gratitude for the support extended by **DR. G. VISHNU MURTHY**, Dean, Department of Computer Science and Engineering, Anurag University. We also express our deep sense of gratitude to **Dr. V. V. S. S. S. BALARAM**, Academic coordinator. **Dr. PALLAM RAVI**, Project Coordinator and project review committee members, whose research expertise and commitment to the highest standards continuously motivated us during the crucial stages of our project work.

We would like to express our special thanks to **Dr. V. VIJAYA KUMAR,** Dean School of Engineering, Anurag University, for their encouragement and timely support in our B. Tech program.

M. Uday Kiran
(20EG105733)
S. Santosh
(20EG105560)
D. Reshwanth
(20EG105516)

ABSTRACT

Dashboards allow managers in keep track of various activities and developments in their departments .There are various existing enterprise resource planning systems that are standardized that can be used by trading sales industry. However there are various modules that are redundant to this industry that affect the speed and efficiency of the system. The proposed system is a statistical dashboard that not only keeps track of all the operations from client request to order delivery but also takes into consideration security of client data by enforcing various security mechanisms. The system has six modules including login module and five dashboards. The login module is secured usingSHA256 hash to store password, variable salt and ReCaptcha mechanism. The five dashboards included in the system are Sourcing Dashboard, Accounts Dashboard, Operation Dashboard, Sales Dashboard and Admin Dashboard. The dashboard is tested to be secured.

Keywords: statistical dashboard; web application, web attacks, mitigation technique

Index

List of Figures

List of Tables

1. Introduction

Admin dashboards offer a quick snapshot of key metrics and performance indicators. Utilizing data visualizations and real-time updates, decision-makers gain actionable insights. Customization and data integration provide a comprehensive view, while interactivity enhances decision support. Crucially, robust security measures safeguard sensitive information on these dashboards.

Purpose:

Central hub for monitoring and managing organizational operations.

Data Visualization:

Visual representation of key metrics using charts and graphs.

KPIs:

Focus on Key Performance Indicators for quick assessment.

Real-time Monitoring:

Provides real-time updates for proactive decision-making.

Customization:

Customizable dashboards to meet user-specific needs.

Interactivity:

Interactive features for detailed data exploration.

Data Integration:

Aggregates data from various sources for a comprehensive view.

Decision Support:

Offers actionable insights for informed decision-making.

Security:

Robust security measures to protect sensitive information.

1.1. Brief Overview of Work

The current organizational setup faces challenges due to the absence of an efficient Admin Dashboard, leading to issues like data overload, inefficient user management, limited analytics, and poor system monitoring. These challenges impact compliance, security, customization, and integration. Addressing these issues is crucial for enhancing operational efficiency, enabling better decision-making, and ensuring a secure organizational environment. Implementing a

comprehensive Admin Dashboard is essential for streamlined management and monitoring, providing actionable insights to administrators.

Disadvantages**:**

- **Overreliance on Metrics:** Narrow focus, neglecting contextual factors; decisions may lack nuance.
- **Data Overload:** Excessive metrics overwhelm users, causing confusion and decision paralysis.
- **Security Concerns**: Displaying sensitive data poses risks, potentially leading to breaches and consequences.

Challenges of Admin Dashboard Analysis:

1. **Data Integration and Quality Assurance:**
    - Diverse data sources necessitate complex integration processes.
    - Ensuring data accuracy and consistency is crucial for reliable insights.
2. **User Interface Design:**
    - Designing intuitive and user-friendly interfaces is challenging.
    - Presenting complex data in an easily digestible format requires expertise.
3. **Selection of Relevant KPIs and Metrics:**
    - Identifying KPIs that align with organizational goals is essential.
    - Metrics must provide actionable insights for decision-making.
4. **Data Security and Privacy:**
    - Maintaining data security while granting appropriate access is critical.
    - Protecting sensitive information is paramount for compliance.
5. **Real-time Data Updates:**

- Keeping the dashboard updated with real-time or near-real-time data is challenging.
- Requires robust infrastructure and efficient data processing capabilities.

6. **Fostering a Data-Driven Culture:**
    - Encouraging administrators to utilize insights requires cultural change.
    - Ongoing training and change management efforts are necessary.

1.2.Objective

The objective of admin dashboard analysis is to harness the power of data to enhance decision-making processes and optimize organizational performance. By extracting meaningful insights from the data presented on the dashboard, this analysis aims to empower administrators with the information they need to make informed decisions that align with the organization's goals. Through the analysis of key metrics, performance indicators, and trends, the objective is to identify areas of improvement, monitor performance, and allocate resources effectively. Additionally, admin dashboard analysis supports strategic planning efforts by providing valuable insights into historical performance and forecasting future trends. Ultimately, the goal is to foster a culture of data-driven decision-making, enhance communication and transparency, and drive continuous improvement throughout the organization.

1.3.Scope

The scope of admin dashboard analysis encompasses the comprehensive utilization of data to drive organizational efficiency and decision-making. It involves the collection and integration of data from diverse sources, followed by the visualization and reporting of key insights through user-friendly dashboards. Performance monitoring and trend analysis are central components, allowing for the tracking of key performance indicators and the identification of historical trends to inform future strategies. Predictive analytics further enhances decision-making by forecasting potential outcomes based on past data. Continuous improvement

initiatives are driven by the identification of optimization opportunities, supported by actionable insights derived from dashboard analysis. Additionally, ensuring data security, compliance, and seamless integration with organizational systems are essential aspects within the scope of admin dashboard analysis, facilitating data-driven decision-making and organizational success.

1.4. Problem Definition

The problem definition for admin dashboard analysis revolves around identifying and resolving inefficiencies in the existing dashboard system that hinder effective decision-making and organizational performance. This entails addressing issues such as fragmented data silos, unintuitive dashboard design impairing data interpretation, and unclear alignment between key performance indicators (KPIs) and organizational objectives. Additionally, concerns related to data security vulnerabilities, delays in data updates, and limited predictive capabilities pose significant challenges. By clearly defining these issues, stakeholders can understand the scope and objectives of the analysis, enabling the development of targeted strategies to enhance data-driven decision-making and organizational efficiency.

1.5. Problem Illustration

Imagine an admin dashboard with multiple data streams flowing into it from various departments within an organization. However, the dashboard appears cluttered and lacks clear organization, making it difficult for administrators to extract meaningful insights. Additionally, data integration issues result in delays and inaccuracies, hindering timely decision-making. Users struggle to navigate the interface, leading to low engagement and underutilization of valuable data resources. Despite the wealth of information available, administrators find it challenging to identify actionable insights amidst the noise. This scenario illustrates the problem in admin dashboard analysis: a disconnect between data availability and usability, impeding the organization's ability to leverage data effectively for informed decision-making and strategic planning.

1.6. Objective of The Project

The objective of the project "admin dashboard analysis" is to evaluate the effectiveness and usability of the existing administrative dashboard within an organization. This involves assessing various aspects such as data integration, visualization, user interface design, and functionality. The primary goal is to identify strengths, weaknesses,

opportunities, and threats associated with the dashboard, aiming to enhance decision-making processes and optimize organizational performance. By analyzing key metrics, performance indicators, and user feedback, the project seeks to provide actionable insights and recommendations for improving the dashboard's design, functionality, and overall usability. Ultimately, the objective is to empower administrators with a user-friendly and data-driven tool that facilitates informed decision-making and supports organizational objectives.

2. Literature Survey

In survey paper “Security analysis of salt/ password hashes” by Praveen Gauravaram, compares different hashing techniques If security is not much an issue we can choose fast hashing algorithms. Fast hashing algorithms can be easily cracked. So chooses low algorithm so that cracker needs more time for creating

Rainbow table .“Dashboard by-Example: A Hypergraph-basedapproach to On-demand Data warehousing systems” by Duong Thi Anh Hoang, Thanh Binh Nguyen, A Min Tjoa In this paper Dashboard-by-Example (DBE) framework andhyperlink based techniques are used to get knowledge fromthe heterogeneous data. Multidimensional hypergraphprovides more flexibility for the changing user requirementwhich helps in dynamic analysis [4]."Study of Efficient and Effective security modelfor database specially designed to avoid internal threats" byAditya A. Shastri and Dr.P.N. Chatur. It mainly focuses onthreats which arise from within the organization. No user isable to obtain information without authorization. Noconcept of encryption is used anywhere in the model ."Detection of session hijacking Vol. Jan 2016 "byJerry Louis. The Session Hijacking has always been themost efficient attack which makes it even harder for thedevelopers to create a secure authentication process.Analyzing the abrupt changes in the database which userdoesn’t remember doing, one can detect the attack. Or avoidusing unsecured Wi-Fi .Based on the survey ERP system was developed taking into consideration customized requirements of thecompany with additional

Theoretical Framework: Usability Construct of System Acceptability Model

Usability has become one of the most vital subjects of both HCI research and practice because it denotes a desired quality of interactive systems and/or products (Tractinsky, 2018). However, to assess the benefit of usability to HCI, the meaning of this concept should be well understood (Hartson & Pyla, 2012). The International Organization for Standardization (9241- 11) defines usability as “the extent to which a product can be used by specified users to achieve specific goals with effectiveness, efficiency and satisfaction in a specified context of use” (Barnum, 2011, p. 11). Since usability draws from different properties of the product to be tested, it serves as an umbrella construct and a building block for developing a testable theory on usability (Stage, 2018; Tractinsky, 2018). Additionally, this construct serves as a basis for scientific or practical measurements of a system or a product (Tractinsky, 2018).

However, the concept of usability alone cannot serve as a tenet that can form a foundation of the usability theory (Stage, 2018). This is because despite its intensive use in HCI research over many years, the potential of usability to form a theory is not well examined (Stage, 2018; Tractinsky, 2018). In the field of human-computer interaction, the practical acceptability of any system, especially its usability, is very important (Nielsen, 1993; Stage, 2018). Thus, this model fits this study because, from the perspective of usability engineering, the usability attribute of the model is vital as it combines diverse methods (easiness, efficiency, memorability, and satisfaction) that are designed to improve the design, development, and evaluation of systems (Nielsen, 1993). In the present study, questionnaires used in previous research were reviewed to suggest criteria for dashboard evaluation. Generally, questionnaires are the most commonly used tools for usability evaluation because of the simplicity of data analysis . According to the findings, although SUS does not cover the efficiency, memorability, or error criteria and consists of a series of general questions for usability evaluation , it was the most widely used tool for dashboard evaluation. In four studies, SUS was used along with other questionnaires for dashboard evaluation .

In the study of Hajesmaeel-Gohari et al., the SUS questionnaire was the most used tool for measuring usability [56]. In the study of Sousa and Dunn Lopez conducted with the aim of identifying the questionnaires used for usability evaluation of electronic health tools, the main used criteria in the investigated questionnaires included learnability, efficiency, and satisfaction. The memorability was the least used criterion .

In the present study, “satisfaction” and “learnability” were proposed as two key criteria for evaluating the usability of the dashboards, and “efficiency” was also proposed as one of the subcriteria of “usefulness.” One criterion, i.e., “memorability,” was not included in the proposed framework, as the learnability could cover the required metrics.

To take advantage of usability evaluation tools, it is important to pay attention to the study objectives, used technologies, and context of use . The ISO/IEC 25010 consists of suitability for tasks, learnability, operability, user error protection, user interface aesthetics, and accessibility . The ISO/IEC 9241-11 also suggests measure such as effectiveness, efficiency, and satisfaction for usability evaluation . Additionally, Nielsen’s criteria were used for evaluating dashboard including efficiency, memorability, error, learnability, and satisfaction . In the current study, usefulness was

used rather than the effectiveness and efficiency criterion, and it was used in four questionnaires, including the Health-ITUES, PSSUQ, CSUQ, and TAM.

In general, TAM and UTAUT are the most widely used acceptance models in health informatics because of their simplicity, and these mainly focus on the usefulness and easy to use technology .

The dashboard “operability” criterion in the current study refers to the user’s ability to the user’s control over the software, error correction ability, and quick recovery. In addition, in previous studies, the “operability” criterion referred to error correction, error correction in use, default value availability in use, message understandability, self-explanatory error messages, operational error recoverability in use, and time between human error operation in use . Moreover, improvement of situational awareness was considered as one of the evaluation criteria for dashboards. Overall, dashboards provide key data that should be monitored effectively to be notified of what is occurring in one’s work environment. The results of previous studies indicated that dashboards have the potential to accelerate data collection, decrease the cognitive load, reduce errors, and improve situational awareness in healthcare settings .

Additionally, the “user interface” criterion includes what a user uses to interact with the system. Some interface hardware components include a keyboard, mouse, microphone, and user interface (e.g., graphic forms, language tools, and interactive tools) . With respect to the user interface of dashboards, the application of visual and interactive features was suggested in the present study, considering data representation and interactive visualization as critical features . Visualization systems, such as dashboards, are capable of two main functions: representation and interaction . Besides interactive features, it is also essential to consider the visual features for an effective and understandable representation of indicators, which can lead to an effective interaction with data and instantaneous monitoring of performance indices In Shneiderman’s study, interactive features included overview, zoom, filter, details-on-demand, relate, history, and extraction . In addition, interactive techniques in M. Khan and S. Khan’s study included zoom and pan, overview and detail, and filtering .

Purpose of the Study and Research Questions

The purpose of the admin dashboard analysis study is to evaluate the effectiveness and usability of the administrative dashboard within the organization.. The following research questions guided this study:

1. How effectively does the current administrative dashboard integrate data from various sources within the organization?
2. What are the strengths and weaknesses of the dashboard's visualization techniques in conveying complex data to users?
3. How intuitive and user-friendly is the dashboard's interface design for administrators?

Researchers' Positionality

Positionality in admin dashboard analysis refers to the researchers' acknowledgment of their own perspectives, biases, and backgrounds that may influence the research process and findings. In this context, researchers should recognize their expertise in data analysis, user experience design, and organizational dynamics, which shape their approach to evaluating the dashboard. Their positionality also encompasses their role within the organization, whether as internal staff or external consultants, influencing their access to data and stakeholders, as well as potential conflicts of interest.

Moreover, researchers should be aware of their own assumptions and preconceptions about the dashboard's effectiveness and usability, as well as any personal or professional interests that may impact their interpretation of the data. Transparency about their positionality is crucial for maintaining objectivity and credibility in the analysis process. By acknowledging their own perspectives and biases, researchers can take steps to mitigate their influence on the research findings and ensure that the analysis remains impartial and objective. This might involve employing diverse research methods, triangulating data sources, and soliciting feedback from multiple stakeholders to corroborate findings and enhance the validity of the analysis. Ultimately, researchers' positionality should be carefully considered throughout the admin dashboard analysis to ensure that the resulting insights accurately reflect the reality of the dashboard's effectiveness and usability within the organization.

Table 2.1. Comparison of Existing Methods

Sl.No	Author	Strategies	Advantages	Disadvantages
1.	Praveen Gauravaram	compares different hashing techniques	If security is not much an issue we can choose fast hashing algorithms	Fast hashing algorithms can be easily cracked
2.	Duong Thi Anh Hoang	framework and hyperlink based techniques	Multidimensional hypergraph helps in dynamic analysis	slow down the loading time of a webpage
3.	Aditya A. Shastri	It mainly focuses on threats which arise from within the organization	No user is able to obtain information without authorization	No concept of encryption is used anywhere in the mode
4.	Jerry Louis	Detection of session hijacking	Analyzing changes in the database which user doesn't remember doing, one can detect the attack.	harder for the developers to create a secure authentication process.

3.PROPOSED METHOD

The existing system was using pen and paper for maintaining the orders. The proposed system will use web technology to design a central hub to manage orders. The project is built in core PHP and MySQL as backend and is hosted on AWS server.

The project contains 6 modules: - Login System, Sourcing Dashboard, Sales Dashboard, Accounts Dashboard, Operations Dashboard and Admin Dashboard. The project is developed in PHP as it is open source, fast, scalable and integrates easily with MySQL. The project also focuses on security aspect, as whole company data is stored in one place, like customer personal details, past orders, margin rates, shipment details, etc.

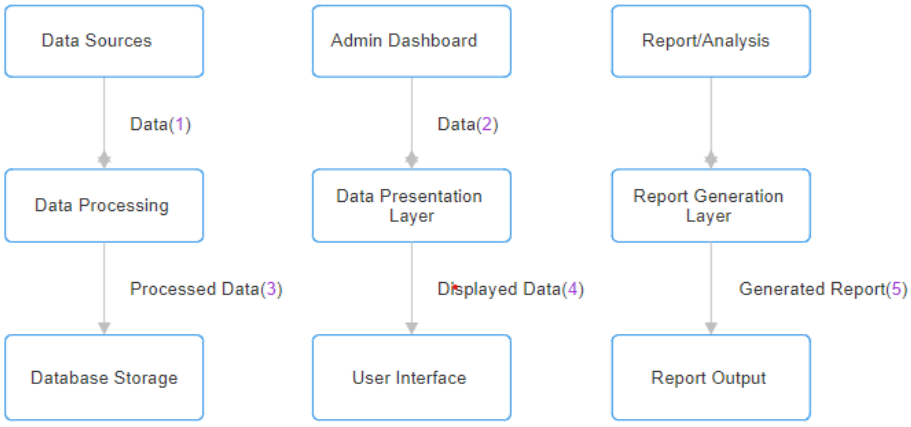


Figure 3.1.1 Concept Tree

3.1 Proposed Method for Admin Dashboard Analysis

1. Requirements Gathering:

- Conduct thorough research to understand the needs and preferences of job seekers and employers.
- Identify key features and functionalities required for the job portal, such as user authentication, job listing management, resume uploading, search and filtering options, and communication tools.

2. Qualitative Methods:

- Conduct user interviews and focus groups to gather insights on user experiences.
- Utilize usability testing to identify pain points and areas for improvement in dashboard interaction.

3. Quantitative Analysis:

- Analyze key performance indicators (KPIs) and usage metrics to assess dashboard effectiveness.
- Examine data trends and patterns through data visualization techniques to identify actionable insights.

4. Triangulation of Findings:

- Combine qualitative and quantitative data to provide a comprehensive understanding of dashboard usability.
- Triangulate user feedback with objective data analysis to validate findings and ensure reliability.

5. Actionable Recommendations:

- Develop recommendations based on identified strengths, weaknesses, and opportunities for improvement.
- Provide actionable insights to enhance dashboard functionality, user experience, and organizational performance.

6. Iterative Process:

- Implement iterative testing and refinement cycles to continuously improve the dashboard.
- Solicit feedback from users and stakeholders to inform ongoing iterations and updates.

7. Documentation and Reporting:

- Document findings and recommendations in a comprehensive report.
- Present actionable insights to stakeholders in a clear and accessible manner to guide decision-making.

11. **Long-Term Monitoring and Evaluation:**

- Establish mechanisms for ongoing monitoring and evaluation of dashboard performance.
- Regularly review dashboard usage and effectiveness to identify evolving needs and opportunities for improvement.

12. **Feedback Mechanisms:**

- Implement feedback mechanisms to gather ongoing input from users and stakeholders.
- Use feedback to inform iterative improvements and updates to the dashboard over time.

After preparing the training data, the next step is to upload it using the Files API. This allows the data to be utilized in fine-tuning jobs effectively. Use the following steps:

Step-1: Import OpenAI module.

Step-2: Create an instance of the OpenAI client for interactions.

Step-3: Open the JSON Lines dataset in binary read mode.

Step-4: Use the OpenAI client to create a new file by uploading the opened file.

Step-5: Set the purpose of the file as "fine-tune."

3.1.1 Dashboard Tools and Service Providers :

Gartner, an American information technology research company and renowned for their market reviews, notes that the business intelligence and analytics market is ”undergoing a fundamental shift” , as new products and competitors emerge, while the customers’ needs continuously change. Further, Gartner states that ”now, a wider range of business users are demanding access to interactive styles of analysis and insights , without requiring them to have IT or data science skills” , supporting the rise of data discovery. The emergence of data discovery has revealed the trend towards ease-of-use tools and the consequential differentiation within the BI market. It is only since 2014 that the – previously niche – data discovery vendors are now included in Gartner’s main report, the ’Magic Quadrant for Business Intelligence and Analytics Platform’, which illustrates their newfound market significance and the importance of self-service BI. Further, Gartner lists interactive exploration and analytic dashboards as critical capabilities for their assessment of current BI solutions – benchmarking those tools on their ability to provide effective data discovery dashboards. This lead traditional business intelligence vendors to try ”very hard to meet the needs of the current market’’and spurred the development of data discovery/dashboard features. Yet, their ”offerings have been pale imitations of the successful data discovery specialists” Back in 2006, Stephen Fewalready mentioned the BI providers’ hurried adaptation to the increasing interest in data discovery and dashboard technology: Most BI vendors that

hadn’t already started offering a dashboard product soon began to do so, sometimes by cleverly changing the name of an existing product, sometimes by quickly purchasing the rights to an existing product from a smaller vendor, and sometimes by cobbling together pieces of products that already existed. Despite data discovery still being in its ”early stages” today’s business intelligence market offers a wide range of data discovery/dashboard tools and providers to choose from Gartner’s Magic Quadrant depicts the current strategic positions of various BI vendors and illustrates the vast amount – but also spread – of BI providers. While this market segment is dominated by emerging, specialised providers, such as Tableau; all leading, conventional BI vendors have released data visualization modules, with examples including IBM Cognos Insight, Microsoft Power View, SAS Visual Analytics, and SAP Crystal Dashboard . It has to be noted that due to a lack of academic BI vendor analyses, market reviews of IT research companies have to be consulted, which may be biased, as many of those companies hold business relations to BI vendors. Yet, various reports agree on the same data discovery tools to be the best-known and most widespread: Qlik and Tableau

Figure 3.1.2. Gartner’s Magic Quadrant for BI and analytics platforms

As the "market leader in data discovery," Qlik was established in 1993 and was recognised as "one of the first large business intelligence and data visualisation software companies on the market" . But Tableau, which was established in 2003, is now the self-service dashboard industry's "gold standard" [17] and has had a big impact on what customers demand from their dashboards in terms of interactivity, appearance, and intuitiveness. Even Nevertheless, there is constant differentiation in the dashboard market as new players like Hadoop big data visualisation start-up Datameer and cloud-based dashboard provider Chartio aim to make an impression by offering a high level

of expertise. The majority of data discovery providers are restricted to licencing and subscription models, even though they provide a variety of deployment choices, including desktop programmes, Web servers, and Software as a Service. Just that The most recent tool from Qlik, Qlik Sense Desktop, has limited features but is free for personal use. Furthermore, the data discovery tools' deployment choices permit the creation of dashboards within the company; they do not, however, permit the use of dashboards to provide information to a client or third party (see, for example, the instance of WINGX Advance GmbH). In these situations, dashboards and the underlying technology need to be customised to order. Yet, this may change as dashboard components are emerging as out-of-the-box libraries and frameworks for Web development.

3.1.2 Dashboard Components for Web Development:

As mentioned in Section 2.2, the term dashboard is generally used with varying interpretations. In current Web development, however, dashboard is understood as the presentation and arrangement of multiple widgets, varying in size and type (e.g. charts, images, text), rather than that of data discovery. There are numerous frameworks that provide dashboard features (see Table 2.1). Yet, most libraries focus on arranging, adding, removing, and resizing tiles. These placeholders are then to be filled by Web developers with individual content (i.e. widgets). While some of the listed dashboard frameworks (e.g. RazorFlow) do provide integrated charting and rudimentary drill-down capabilities, none can be considered a framework for data discovery dashboards (i.e. providing brushing and linking). Despite the – current – unavailability of a full-stack discovery dashboard framework, some libraries do facilitate their development. Charting, the fundamental element of any discovery dashboard, is being provided by countless libraries – so many in fact, that Table 2.2 is merely an excerpt of JavaScript libraries that provide charting with interactive elements (e.g. mouse-over events). The restriction on JavaScript stems from its dominating market share for client-side languages used for Web pages (90% as of August 2015) [23]. To be emphasised, first of all, are VanCharts, ZingChart, and D3.js. In contrast to most other pure charting libraries, VanCharts and ZingChart also offer integrated chart arrangement capabilities

and support functions upon selecting a data item within a chart – enabling

Name	Characteristic(s)	License
angular-dashboard-framework	Open-source, module for Angular.js	MIT
Atlasboard	Open-source, module for Node.js	Apache
Dashing	Open-source, Ruby gem	MIT
FnordMetric	Open-source, for charting using SQL	GPL
Gridster.js	Open-source, jQuery plugin	MIT
jDash	For Asp.Net users	Commercial
jSlate	Provided as service	Free
Pyxley	Open-source, for Python users	MIT
RazorFlow	Open-source, provides rudimentary drill-down	Commercial
Reportr	Open-source, built using Node.js	Apache
Shiny	Open-source, for R users	GPL

Table 3.1.1. List of dashboard/widget-handling frameworks

the development of basic drill-downs. However, brushing and linking are not supported which would lead to dashboards consisting of interactive – yet, isolated – charts. D3.js, on the other hand, stands out as it provides a thorough basis for all sorts of interactive data visualizations and, thus, quickly gained in popularity after its release in 2011 [24, 25]. By now, D3.js is used to power visualizations across industries (for example, Datameer also uses D3.js [26]) and has already brought about numerous instruction and guidance books [27, 28, 29]. It is also D3.js' versatility that allowed for Web-based charts providing brushing and linking [24] and focus+context capabilities. Further, numerous charting libraries were released building on D3.js, which provide preconfigured charts or functions. C3.js, for example, provides all basic chart types (such as bar-, line, and pie-charts) in an easy-to-use manner while requiring only minimal programming or D3.js knowledge. Yet, C3.js – as most other charting libraries – supports neither brushing nor linking. Crossfilter, a JavaScript library for exploring multidimensional data [30, 31], was released in 2012 under Apache license and facilitates coordinated views. Crossfilter initiated the development of dc.js, a charting library combining Crossfilter and D3.js to allow brushing and linking across multiple charts. Still, filtered data context can only be retained within singular views with dc.js. No currently available library facilitates brushing, linking, and focus+context across multiple or all graphs of a Web-based dashboard. Nonetheless, current libraries and frameworks can be utilised to build a coherent discovery dashboard – using solely free

and open-source resources. The development of such a discovery dashboard is intended to be a major research outcome of this project – as stated in Section 1.2. Further, the development of the exemplary dashboard shall reveal possibilities for advancing existing libraries and spur the development of dashboards with data discovery capabilities

3.2 Dashboard Architecture:

”The architecture of performance dashboards”, explains Eckerson,”have followed the trajectory of software architectures in general, from mainframe computing to client/server computing to Web-based architectures”. More important than choosing a specific technology is understanding how and where the user interface, application logic, and data processing are being handled . Hence, Eckerson implies to pursue a Model-View-Controller architecture , which ”has been widely embraced as an approach for developing Web-based applications” and has shown to increase maintainability and re-usability by resulting in less coupled, more cohesive architectures . With the emergence of Web applications, processing shifted from desktop machines (i.e. fat clients) to application servers where desktop machines only render the HTML-based representation within a Web browser (i.e. thin clients) [9, p. . According to Eckerson , this has the advantage of avoiding client-side processing loads and local software installation. Further benefits are the possibility of a centralised administration and increased security . However, ”the downside of HTML-based thin clients is lack of performance and functionality” , Eckerson notes; as client-server communication is prone to latency and, thus, introduces a delay between user action and system response. On this basis, Web-based architectures experience a ’thickening’ as to ”take advantage of the processing power of desktop computers and make Web-based applications more interactive and dynamic” resulting in, so-called rich Internet applications. Rich Internet applications can be realised using various Web-based technologies, such as Java applets or ActiveX controls, embedded scripting languages (e.g. JavaScript, TypeScript), or multimedia frameworks (e.g. Adobe Flash, Microsoft Silverlight). As Java applets and ActiveX controls represent ”mini-applications that run inside a Web browser and execute within a virtual machine” , they more closely resemble desktop applications than Web pages. Further, these software framework are infamous for raising security concerns and are gradually reaching the end of their useful life, as HTML5 replaces them in many of their functions. Embedding scripting languages inside HTML pages,

on the other hand, is a more lightweight approach . Via scripts, a developer can modify the downloaded HTML page or ”[retrieve] new content from the server [...] without interfering with the display and 11 behaviour of the page”– so-called asynchronous JavaScript and XML (AJAX). Currently, Web developers’ most prevalent choice, to achieve dynamic and interactive Web pages, is using scripting languages – in particular, JavaScript . Yet, Eckerson notes that using scripting languages also has its disadvantages, as one has to ensure crossbrowser compatibility and may potentially experience performance and reliability issues. ”Another popular approach”, Eckerson adds, ”is to use multimedia development platforms, such as Adobe Flash”. These multimedia frameworks, however, require the user to ”download a Web browser plugin , which remains permanently installed on their machine” . Further, Eckerson puts forth the increasing browser support for multimedia plugins, their browser-independence, and offline capabilities as advantages. However, his book was published in 2010 and Web technology has changed remarkably since then. HTML5 has rendered most plugins obsolete by providing native multimedia capabilities and was officially recommended by the World Wide Web Consortium (W3C) in 2014 . While Eckerson hints at HTML5, he could not foresee the shift towards ”plugin-free browsing” as plugins – executing outside of the browser – require browserindependent updating and are often prone to vulnerabilities . Thus, the combination of HTML5 and JavaScript has established itself as the current standard for interactive Web development, with ”no end in sight to the rise of JavaScript” especially, with its ”proliferation of open source tools and libraries” . In contrast to commercial data discovery products, which make use of complex, multilayered architecture and deployment structures (see for example Qlik and Tableau architectures); a purely Web- and open-source-based discovery dashboard will have its architecture somewhat defined by the available and utilised libraries. Crossfilter , the earlier-mentioned JavaScript library facilitating in-browser cross-dimensional data analysis, for example, was developed to execute on client-side – burdening the local Web browser with all of a dashboard’s calculations. However, there are a few exploratory attempts at utilising Crossfilter on server-side (i.e. on a Web server) with the aim of providing better support for massive datasets, which cannot be processed within the memory limits of a Web browser . Besides influencing the requirements on client-side hardware, the decision between client- and server-side processing is based on the trade-off between the introduction of a delay (due to necessary client-server communication) and decreased processing time (as a server is

expected to provide more processing power). Regardless of a particular architecture, all dashboards are subject to common HCI rules and principles and are, thus, supposed to adhere to existing guidance for the front-end design of dashboards. It is Eckerson, who concludes that ”dashboard architects need to consider various Web technologies to deliver an attractive, interactive, and high-performance user interface” .

3.2.1 Dashboard Design:

By definition, dashboards greatly rely on visualisations. The interface – the look and feel of a dashboard – can determine whether it succeeds or fails, according to Eckerson . ”Information presentation is a balancing act” between conveying a lot of information and not overwhelming the user; capturing users’ attention and not distracting them; and between making it feel intuitive while still offering a wide range of features. To tackle the challenge of developing a well-designed dashboard, one can subdivide the topic into four major parts: form, structure, design principles, and functionality

Form:

Technically, a dashboard’s form – i.e. the method of delivering a dashboard – can range from Microsoft Excel sheets and online apps to static screens or even paper-based presentations . The decision for or against a certain form should be made based on a dashboard’s requirements, such as timeliness, mobility, detail, or interactivity . This project, however, will focus exclusively on the delivery of an online application – which, is described as being the most versatile of all dashboard forms . While there are differing opinions ,most authors suggest confining a dashboard to a single screen . ”A dashboard is meant to be viewed at-a-glance”, explain Lisa Pappas and Lisa Whitman , ” without having to scroll or navigate to multiple pages”. They reason that ”this allows for processing the information with minimal effort” . Wayne Eckerson agrees and also states that ”users should not have to scroll down or across a screen to view critical data” . He admits, though, that presenting all relevant data within such constraints is the ”first and toughest goal of a dashboard designer” . Stephen Few puts forth the limitations of our short-term memories as a reason for ensuring that a dashboard stays confined to a single screen. Few further elaborates: One of the great benefits of a dashboard as a medium of communication is the simultaneity of vision that it offers: the ability to see

everything that you need at once. This enables comparisons that lead to insights, those ”Aha!” experiences that might not occur in any other way. Clearly, exceeding the boundaries of a single screen negates this benefit. To achieve single-page – yet, highly informative – dashboards, developers must make use of space-efficient visualisations , rigorously exclude non-essential information , and utilise dashboard structure to convey meaning

Structure:

”Dashboard content must be organized in a way that reflects the nature of the information and that supports efficient and meaningful monitoring”, explains Stephen Few , when he emphasizes to arrange, position, and size dashboard elements according to their importance. As users pay particular attention to the top left quadrant of a dashboard, while the bottom right quadrant receives least attention; developers must place the most essential dashboard element prominently within the top left part of the screen . Besides positioning, it is also the visual grouping of elements that intuitively conveys relations between dashboard items or indicates a flow of expected actions . Grouping can be achieved by intelligent use of whitespace and adhering to a pre-defined grid layout . Such grid systems, i.e. columns of equal 13 width, ensure consistent alignment of elements and ”brings a coherence and order to the page that puts users at ease” . Figure 3.2.2: Grid layout of New York Times’ Web site (highlights added) White-space, an important aspect in interface design and often overlooked , can be used to group or separate dashboard items without requiring any additional graphical elements (such as background images or borders) .Additionally, by using spacing to indicate inter-element relations, one ”[creates] places for the eye to ’rest’ so that the non white space has more impact” . Without spacing, visual prioritisation would be lost, as elements would seamlessly blend into each other . One can, for example, observe the use of whitespace between the columns of the illustrated and highlighted grid layout as shown in Figure 3.2.2

Figure 3.2.2. Grid layout of New York Times' Web site (highlights added)

Design Principles:

Furthermore, a set of design principles has proven itself to be beneficial with regards to user perception and is, thus, often referred to within dashboard design literature: the socalled 'Gestalt' principles by Moore and Fitz . 'Gestalt', the German word for pattern, describes the psychology behind "the way our minds perceive wholes out of incomplete elements" and offers insight that can directly be applied to dashboard design . Among these principles – and of particular interest for dashboard design – are proximity, similarity, closure, continuity, symmetry, and figure & ground Proximity All else being equal, elements that are placed closer together than others are perceived to be more related to each other than to distant elements .Similarity Similar elements (via – for example – colour, size, or shape) are perceived to be more related to each other than to dissimilar elements .Closure When an element is incomplete or a space is not completely enclosed, people tend to see a recognizable pattern by filling in missing information .Continuity People tend to perceive intersecting elements as independent, uninterrupted elements .Symmetry Symmetrical elements are perceived to more related to each other than unsymmetrical elements and centre around a focus point . Figure & ground The eye differentiates an element (i.e. figure) form its surrounding area (i.e. background). Balancing figure and ground improves the clarity of a perceived image . In order to maintain a pleasing and clear dashboard appearance, a designer should minimise decorations and not misuse or overuse colour . "Variations in chart color that do not encode a meaning can be another source of distraction or distortion"

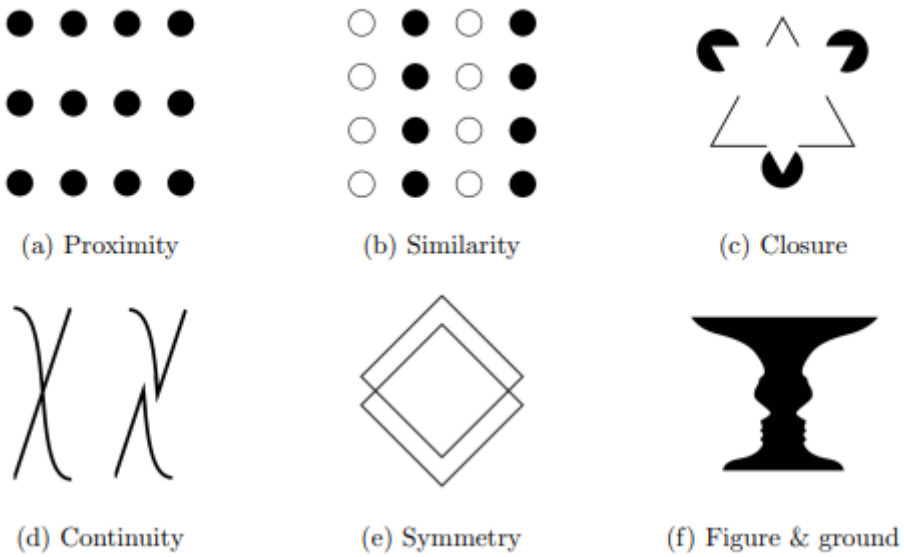


Figure 3.2.3. Combining the summaries

4. Implementation

4.1 Functionality:

Implementing admin dashboard analysis using the MERN (MongoDB, Express.js, React.js, Node.js) stack involves several steps. Here's a general outline:

4.1.1 Set Up MongoDB Database:

- Install MongoDB on your system or use a cloud-based MongoDB service like MongoDB Atlas.
- Create a new database and define the schema for storing dashboard data.

4.1.2 Create Backend with Node.js and Express.js:

- Set up a Node.js project.
- Install Express.js for creating the backend server.
- Define API routes for CRUD (Create, Read, Update, Delete) operations to interact with the database.
- Implement authentication and authorization mechanisms to secure the dashboard data.

4.1.3 Integrate MongoDB with Node.js:

- Use a MongoDB driver like Mongoose to connect Node.js with MongoDB.
- Define models representing data entities in the database and establish relationships between them.

4.1.4 Develop Frontend with React.js:

- Set up a React.js project using create-react-app or a similar tool.
- Design the dashboard layout and components using libraries like Material-UI or Bootstrap.
- Implement data fetching from the backend APIs using tools like Axios or Fetch API.
- Create interactive data visualizations using charting libraries like Chart.js or D3.js.

4.1.5 Connect Backend and Frontend:

- Configure CORS (Cross-Origin Resource Sharing) to allow frontend access to backend APIs.
- Make API requests from the frontend to fetch data from the backend and display it on the dashboard.

4.1.6 Implement Dashboard Features:

- Add features such as filtering, sorting, and searching to allow users to interact with the data.
- Implement real-time updates using WebSocket or Server-Sent Events for live data streaming.
- Create user authentication and authorization mechanisms to control access to the dashboard.

4.1.7 Testing and Deployment:

- Test the dashboard thoroughly to ensure functionality and usability.
- Deploy the backend on a server using platforms like Heroku or AWS Elastic Beanstalk.
- Deploy the frontend on a static file server or utilize platforms like Netlify or Vercel for deployment.
- Set up continuous integration and continuous deployment (CI/CD) pipelines for automated testing and deployment.

4.1.8 Monitoring and Maintenance:

- Set up monitoring tools to track server performance, errors, and usage metrics.
- Monitor database performance and optimize queries for better efficiency.
- Regularly update dependencies and security patches to ensure the dashboard remains secure and up-to-date.

By following these steps, you can implement an admin dashboard analysis application using the MERN stack, providing users with a powerful tool for visualizing and analyzing organizational data.

4.2 Attributes:

The attributes you choose for your admin dashboard analysis will depend on the specific purpose of your dashboard and the type of data you're working with. However, here are some general categories of attributes to consider:

4.2.1 User Management (if applicable):

- **User:** Username, email, role (admin, editor, viewer), department, location (optional).
- **Activity:** Login time/date, last activity time/date, actions performed (e.g., created report, edited data).

4.2.2 System Health:

- **Component:** Server name, service name, status (running, stopped, error).
- **Performance:** Resource utilization (CPU, memory, disk), response times, error rates.
- **Logs:** Number of errors/warnings, severity level, timestamp.

4.2.3 Content Management (if applicable):

- **Content:** Title, type (article, page, product), creation date, author, last modified date.
- **Performance:** Views, comments, shares (for social media integration).
- **Engagement:** Average time spent on content, completion rate (for forms).

4.2.4E-commerce (if applicable):

- **Product:** Product name, category, price, stock level, average rating.
- **Order:** Order ID, customer details, order date, status (placed, shipped, delivered, canceled), total amount.
- **Sales:** Revenue by product or category, number of orders, conversion rate.

4.2.5 General Data Analysis:

- **Time:** Timestamp, date, hour, day of week, month, year.

- **Location:** Country, region, city (if relevant to your data).
- **User Segment:** Demographics (age, gender), device type, operating system (for website analytics).
- **Metrics:** Number of users, sessions, actions (e.g., clicks, downloads), conversion rate (if applicable).

4.2.6 Additional Tips:

- **Focus on Key Performance Indicators (KPIs):** Identify the most important metrics that provide insights into the success of your system, content, or overall goals.
- **Filter by Relevant Attributes:** Allow users to filter data based on specific attributes to focus on areas of interest.
- **Drill Down for Details:** Enable users to click on chart elements or data points to see more detailed information.
- **Visualize Effectively:** Choose appropriate chart types (bar charts for comparisons, line charts for trends) to effectively represent your data.
- **Consider Real-Time Data:** If applicable, integrate real-time updates for critical metrics to provide administrators with up-to-date information.

By selecting relevant attributes and using them effectively in your dashboard design, you can create a valuable tool for admins to monitor system health, user activity, content performance, and other key aspects of your application or service.

4.3. Experiment Screenshots;

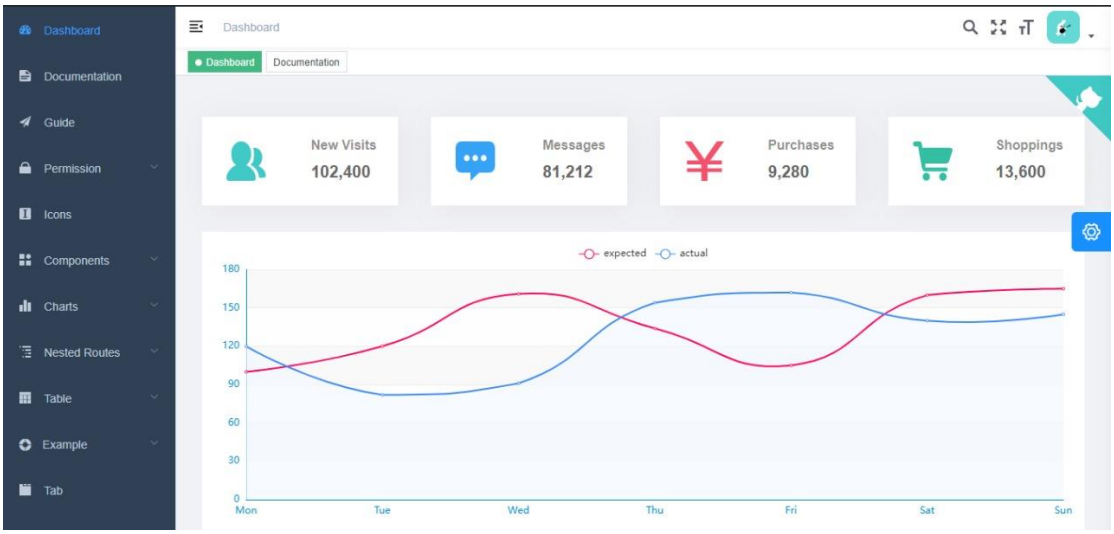


Figure 4.3.1. Main Dashboard

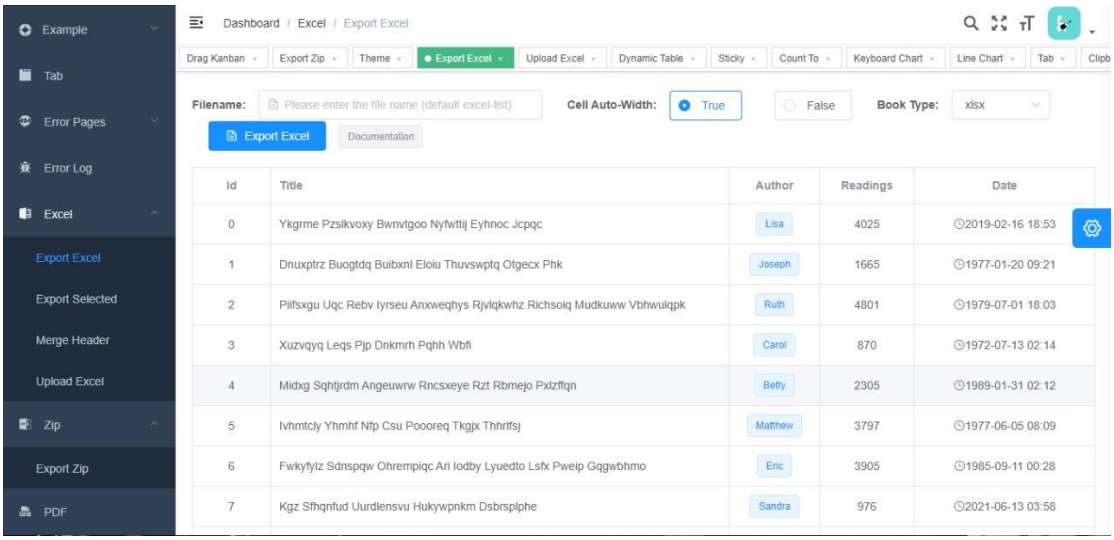


Figure 4.3.2. Excel Export

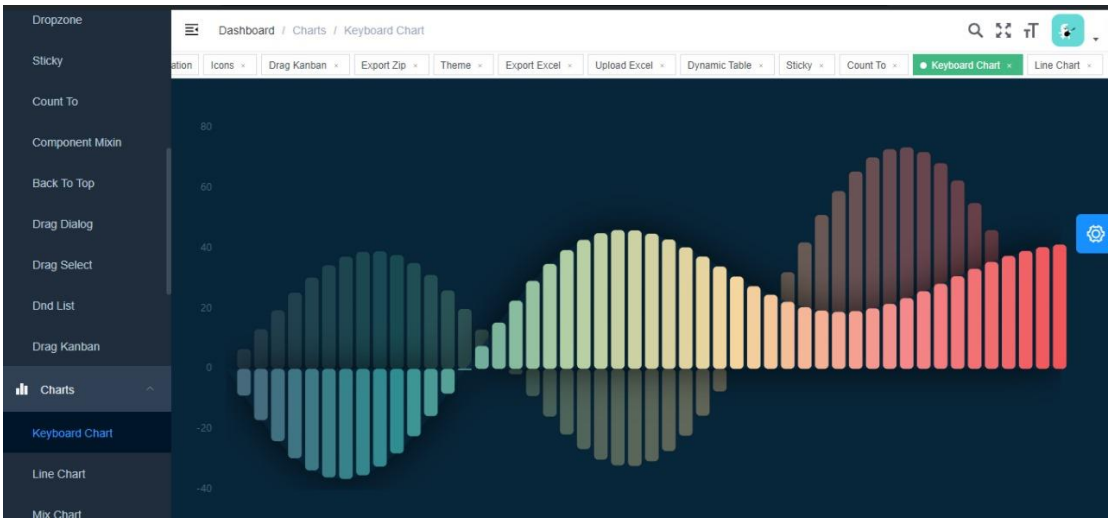


Figure 4.3.2.Graphs for Dashboard

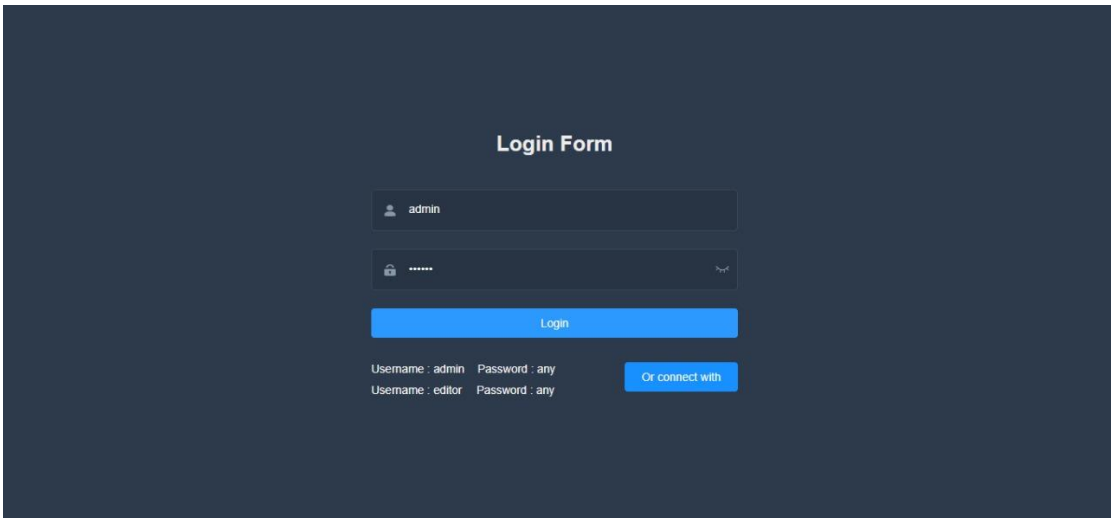


Figure 4.3.3. Admin/User Login page

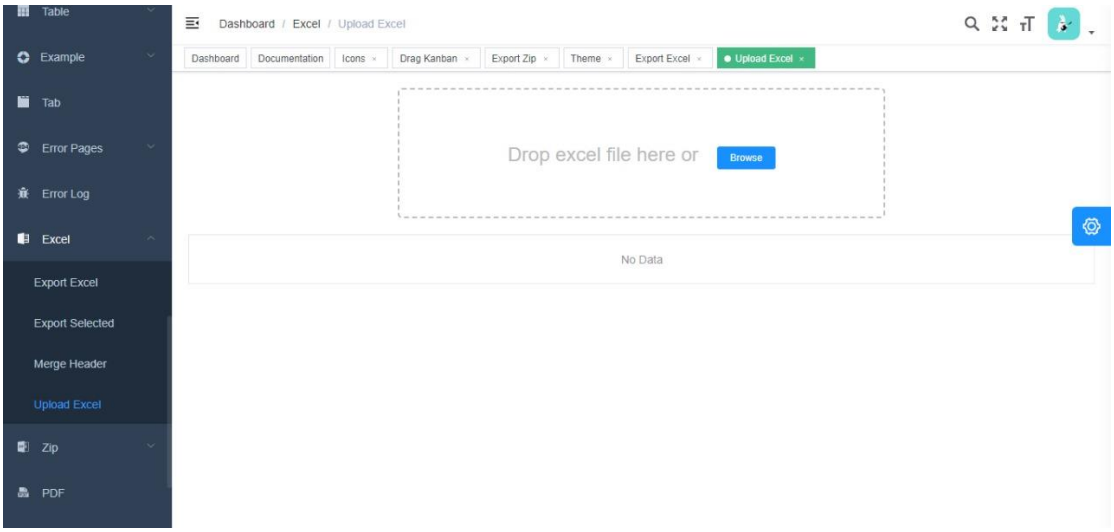


Figure 4.3.4. File Ex

4.4. Dataset:

For admin dashboard analysis, the choice of dataset depends on the specific objectives, industry, and organizational context. However, here are some common types of datasets that can be used for admin dashboard analysis:

Sales and Revenue Data: This dataset includes information about sales transactions, revenue generated, product/service performance, customer demographics, and sales channels. It can be used to analyze sales trends, customer behavior, and the effectiveness of marketing strategies.

Financial Data: Financial datasets encompass various financial metrics such as expenses, profits, cash flow, budget allocations, and financial ratios. Analysis of financial data can provide insights into the organization's financial health, investment decisions, and cost optimization opportunities.

Operational Data: Operational datasets include information about production processes, inventory levels, supply chain logistics, and resource utilization. Analyzing operational data can help identify bottlenecks, streamline processes, and optimize resource allocation.

Customer Data: Customer datasets contain demographic information, purchase history, feedback, and satisfaction scores. Analysis of customer data can inform customer segmentation, retention strategies, and personalized marketing campaigns.

Employee Data: Employee datasets include information about workforce demographics, performance metrics, training records, and employee satisfaction surveys. Analysis of employee data can help optimize workforce management, identify training needs, and improve employee engagement.

Website or App Analytics: Website or app analytics datasets provide information about user behavior, traffic sources, conversion rates, and engagement metrics. Analysis of website or app analytics can inform user experience optimization, content strategy, and digital marketing efforts.

Social Media Data: Social media datasets contain information about brand mentions, sentiment analysis, engagement metrics, and audience demographics. Analysis of social media data can help assess brand reputation, monitor competitor activity, and identify market trends.

Survey Data: Survey datasets include responses to organizational surveys, feedback forms, or market research studies. Analysis of survey data can provide insights into customer preferences, satisfaction levels, and areas for improvement.

It's important to ensure that the chosen dataset is relevant to the objectives of the admin dashboard analysis and complies with data privacy regulations. Additionally, data quality and completeness are crucial considerations to ensure the accuracy and reliability of the analysis results.

SAT Dash	3	Sep 15, 2021	Apr 13, 2023
MusIQ Results	1	Feb 14, 2023	Mar 29, 2023
Engagement 2021	1	Feb 9, 2022	Mar 28, 2023
Restaurant feedback for text analysis	2	Mar 28, 2023	Mar 28, 2023
CSAT Dashboard for Managers	2	Sep 15, 2021	Mar 3, 2023
Mock WAF Data	1	Mar 3, 2023	Mar 3, 2023
Call Center Data 2	1	Mar 2, 2023	Mar 2, 2023

Table 4.4.1. Dataset

5. Experimental Setup

5.1. Data Collection and Preparation:

Identify Data Points: Determine the key metrics you want to analyze within the admin dashboard. Here are some examples:

User Management: Number of registered users (job seekers and employers), user activity (login frequency, searches conducted).

Dashboard Management: Number of user postings, application rates, average number of registrations.

Engagement: Interaction patterns (e.g., job views, application submissions, message exchanges between job seekers and employers).

Data Sources: Integrate your admin dashboard with relevant parts of your MERN stack application and database:

Back-end API endpoints that provide access to user data, job data, and application data.

Consider querying data directly from MongoDB using aggregation pipelines for complex analysis tasks.

Data Transformation (Optional): If needed, pre-process the collected data to prepare it for analysis. This could involve:

Cleaning incomplete or invalid data entries.

Filtering data based on specific timeframes or user roles.

Aggregating data (e.g., total applications per month, average time to application decision).

5.1.1. Admin Dashboard Design:

Visualization Tools: Choose appropriate libraries or frameworks for creating informative visualizations:

Charts (bar graphs, line graphs, pie charts) for displaying trends in user registrations and application rates.

Tables for detailed breakdowns of user activities, application statuses, and other relevant data points.

Interactive elements (e.g., filters, drill-downs) to allow admins to explore data from different perspectives.

User Interface (UI) Design: Create a user-friendly and intuitive UI for the admin dashboard:

- Clear labels and descriptions for each visualization.
- Easy access to filters and sorting options.
- Integration with user profiles or job listings for deeper insights when necessary.
- Front-end Technologies: Use libraries like Chart.js, D3.js, or React libraries like React Chart js 2 for visualization components. Ensure compatibility with your existing React application.

5.2. Analysis and Insights:

Define Goals: Determine what you want to achieve with the data analysis. This might include:

- Understanding user demographics and behavior patterns.
- Identifying trends in job postings and applications.
- Evaluating the effectiveness of the job portal for connecting job seekers and employers.
- Analysis Techniques: Utilize data analysis techniques to extract insights from the visualizations:
- Identify correlations between different data points (e.g., job location and application rates).
- Analyze user behavior patterns (e.g., most popular job search keywords, application completion rates).
- Compare performance metrics over time to track progress and identify areas for improvement.
- Actionable Insights: Translate the insights into actionable steps to enhance your job portal:
- Improve job search functionality and user experience based on user behavior analysis.
- Identify in-demand skills or job categories based on application trends.
- Make data-driven decisions regarding marketing efforts or platform features.

5.2.1. Experimental Design:

Hypotheses: Formulate hypotheses about user behavior or platform effectiveness based on your chosen data points.

A/B Testing (Optional): If feasible, consider implementing A/B testing to compare different versions of UI elements or functionalities within the admin dashboard or the

job portal itself to see which version leads to better outcomes.

Iteration: Continuously iterate on your data collection, analysis methods, and dashboard design based on the insights you gain and evolving needs.

5.2.2. Monitoring and Reporting:

Scheduled Data Refresh: Update the visualizations and data points in the admin dashboard regularly to reflect real-time trends and platform performance.

Automated Reports (Optional): Develop automated reporting features to generate periodic reports (daily, weekly, monthly) with key metrics and visualizations for admin users.

Alerting System (Optional): Implement an alerting system that notifies admins of anomalies or significant changes in key metrics identified through the dashboard.

5.3 Prerequisites:

- Ensure you have Node.js and npm (or yarn) installed on your system. You can verify this by running node -v and npm -v (or yarn -v) in your terminal.

2. Project Initialization:

- Create a new project directory and initialize a Node.js project using npm:

```
Bash

mkdir mern-admin-dashboard
cd mern-admin-dashboard
npm init -y
```

3. Install Dependencies:

- Install the required MERN stack dependencies and additional libraries needed for the backend:

```
Bash

npm install express mongoose cors body-parser
```

Explanation of Dependencies:

- Express: A popular Node.js framework for building web applications and APIs.
- Mongoose: An Object Document Mapper (ODM) for MongoDB, allowing you to interact with the database using JavaScript objects.
- cors: Enables Cross-Origin Resource Sharing (CORS) if your frontend and backend are on different domains.

- Body - parser: A middleware that parses incoming request bodies (usually JSON data) to make them accessible in your backend code.

4. Additional Considerations:

- **Database Connection:** You'll need to configure a connection string to your MongoDB database instance. Replace the placeholder in the following example with your actual connection details:

```javascript
mongoose.connect('mongodb://localhost:27017/your_database_name', {
  useNewUrlParser: true,
  useUnifiedTopology: true
})
```

- **Data Models:** Define your data models using Mongoose schemas to represent the data you'll store in your database (e.g., users, data points for analysis).
- **API Endpoints:** Create API endpoints in your Express app to handle requests from the frontend. These endpoints will be used to fetch and potentially manipulate data for analysis.
- **Error Handling:** Implement proper error handling mechanisms to catch and respond to potential issues during database operations or API requests.

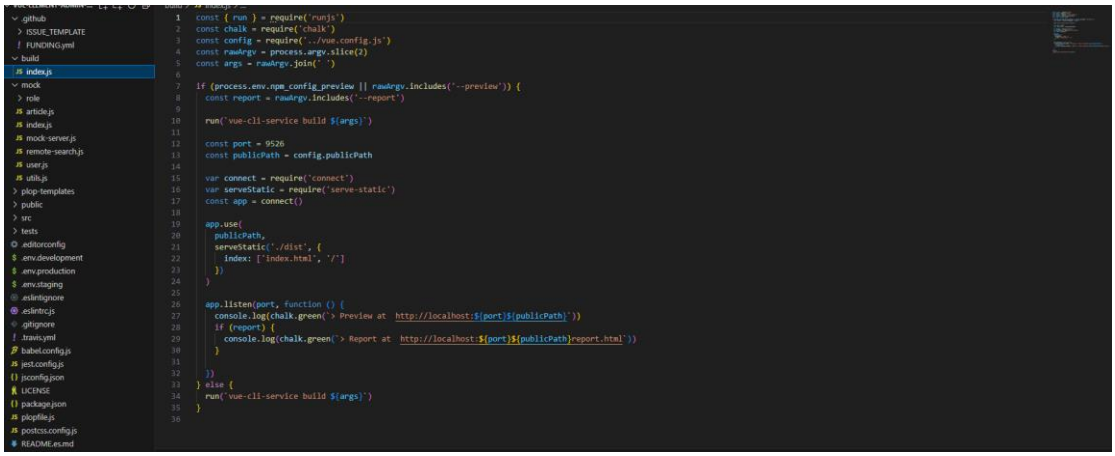
5. Next Steps:

- Follow the next steps to build the frontend (React.js) part of your application, where you'll implement the user interface and connect to the backend API endpoints to fetch and visualize the data for analysis.
- Consider additional libraries for data fetching (e.g., Axios) and state management (e.g., Redux) in your React application.

Remember, these are the basic installation steps for the backend. You'll need to develop the actual logic for data storage, retrieval, and manipulation to suit your specific analysis needs.

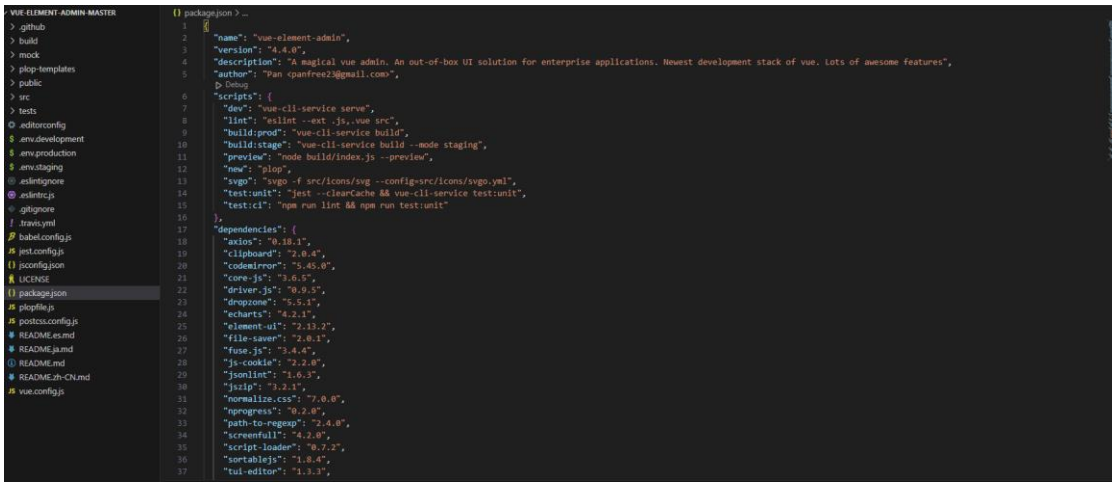
```
const Mock = require('mockjs')

const List = []
const count = 100

const baseContent = '<p>I am testing data, I am testing data.</p><p><img src="https://wpimg.wallstcn.com/4c69009c-0fd4-4153-b112-6cb53d1cf943"></p>'
const image_uri = 'https://wpimg.wallstcn.com/e4558086-631c-425c-9430-56ffb46e70b3'

for (let i = 0; i < count; i++) {
  List.push(Mock.mock({
    id: '@increment',
    timestamp: +Mock.Random.date('T'),
    author: '@first',
    reviewer: '@first',
    title: '@title(5, 10)',
    content_short: 'mock data',
    content: baseContent,
    forecast: '@float(0, 100, 2, 2)',
    importance: '@integer(1, 3)',
    'type|1': ['CN', 'US', 'JP', 'EU'],
    'status|1': ['published', 'draft'],
    display_time: '@datetime',
    comment_disabled: true,
    pageviews: '@integer(300, 5000)',
    image_uri,
    platforms: ['a-platform']
  }))
}

module.exports = [
  {
    url: '/vue-element-admin/article/list',
    type: 'get',
    response: config => {
      const { importance, type, title, page = 1, limit = 20, sort } = config.query

      let mockList = List.filter(item => {
        if (importance && item.importance !== +importance) return false
```

Figure. 5.1. Coding environment screenshot



```
const { run } = require('runjs')
const chalk = require('chalk')
const config = require('../vue.config.js')
const rawArgv = process.argv.slice(2)
const args = rawArgv.join(' ')

if (process.env.npm_config_preview || rawArgv.includes('--preview')) {
  const report = rawArgv.includes('--report')

  run(`vue-cli-service build ${args}`)

  const port = 9526
  const publicPath = config.publicPath

  var connect = require("connect")
  var serveStatic = require("serve-static")
  const app = connect()

  app.use(
    publicPath,
    serveStatic('./dist', {
      index: ['index.html', '/']
    })
  )

  app.listen(port, function () {
    console.log(chalk.green(`> Preview at  http://localhost:${port}${publicPath}`))
    if (report) {
      console.log(chalk.green(`> Report at  http://localhost:${port}${publicPath}report.html`))
    }

  })
} else {
  run(`vue-cli-service build ${args}`)
}
```

Figure. 5.2. Coding environment screenshot



```
{
  "name": "vue-element-admin",
  "version": "4.4.0",
  "description": "A magical vue admin. An out-of-box UI solution for enterprise applications. Newest development stack of vue. Lots of awesome features",
  "author": "Pan <panfree23@gmail.com>",
  Debug
  "scripts": {
    "dev": "vue-cli-service serve",
    "lint": "eslint --ext .js,.vue src",
    "build:prod": "vue-cli-service build",
    "build:stage": "vue-cli-service build --mode staging",
    "preview": "node build/index.js --preview",
    "new": "plop",
    "svgo": "svgo -f src/icons/svg --config=src/icons/svgo.yml",
    "test:unit": "jest --clearCache && vue-cli-service test:unit",
    "test:ci": "npm run lint && npm run test:unit"
  },
  "dependencies": {
    "axios": "0.18.1",
    "clipboard": "2.0.4",
    "codemirror": "5.45.0",
    "core-js": "3.6.5",
    "driver.js": "0.9.5",
    "dropzone": "5.5.1",
    "echarts": "4.2.1",
    "element-ui": "2.13.2",
    "file-saver": "2.0.1",
    "fuse.js": "3.4.4",
    "js-cookie": "2.2.0",
    "jsonlint": "1.6.3",
    "jszip": "3.2.1",
    "normalize.css": "7.0.0",
    "nprogress": "0.2.0",
    "path-to-regexp": "2.4.0",
    "screenfull": "4.2.0",
    "script-loader": "0.7.2",
    "sortablejs": "1.8.4",
    "tui-editor": "1.3.3",
```

Figure. 5.3. Packages

5.4 Parameters:

Data Selection:

- **Metrics:** Identify the Key Performance Indicators (KPIs) that represent the success of your system or content. These are the measurable values you want to track and analyze. Examples include number of users, sales revenue, conversion rate, average time spent on a page, etc.
- **Dimensions:** These are the attributes that categorize or group your data for analysis. They help you understand how your KPIs vary across different segments. Examples include user type (admin, editor, viewer), product category, geographic location, date/time, device type, etc.
- **Filters:** Allow users to filter data based on specific dimensions to focus on areas of interest. This helps them delve deeper into specific aspects of the data.

Data Presentation:

- **Visualization:** Choose appropriate charts and graphs to visually represent your data. Consider factors like comparisons (bar charts), trends (line charts), and distributions (pie charts). Use color and labels effectively for clarity.
- **Data Table:** Optionally, include a data table beneath charts to provide raw data points for reference. This allows users to drill down into specific values.
- **Alerts and Notifications:** Set up alerts or notifications for critical metrics that fall outside predefined thresholds. This helps administrators identify potential issues quickly.

User Interactivity:

- **Drill Down:** Enable users to click on chart elements or data points to see more detailed information about that specific segment. This allows for deeper analysis.
- **Time Range Selection:** Allow users to choose a specific time range for analysis (e.g., daily, weekly, monthly, custom). This provides flexibility for viewing trends and historical data.

- **Customization:** Consider allowing users to customize the dashboard layout and prioritize the metrics they find most important. This caters to individual preferences and workflow needs.

Additional Considerations:

- **Real-Time Data:** If applicable, integrate real-time updates for critical metrics to keep administrators informed of the latest situation.
- **Data Refresh Rate:** Set an appropriate refresh rate for the dashboard data to balance responsiveness with server load.
- **Export Functionality:** Allow users to export data as CSV or other formats for further analysis in external tools.

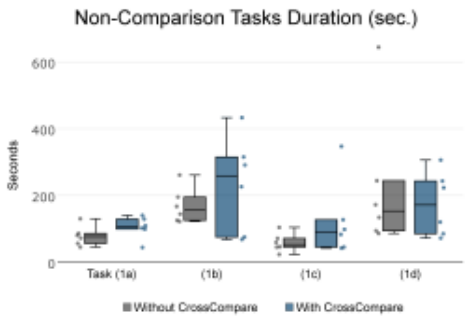
By carefully defining these parameters, you can create an admin dashboard that effectively communicates key insights, empowers administrators to make informed decisions, and improves overall system/content management efficiency.

6. Discussion of Results

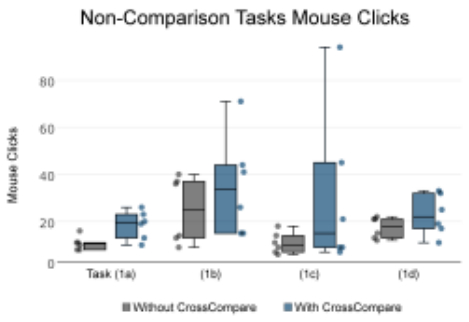
While the identical entry and exit questionnaires allow for a combined examination of the overall perceived usability of the air traffic dashboard, the quantitative analysis will be split according to the first and second study, as the differing task experiment structures render them incomparable.

Non-Comparison Tasks Analysis:

Each task of the first study was performed 12 times, once by each participant, in equal proportions either with or without the comparison feature. Each task has, thus, been performed six times with and six times without Cross Compare. Across all tasks of the first study, the mean (M) task completion time without the comparison feature yielded 133.54 seconds, with a standard deviation (SD) of 125.17 seconds. With the comparison feature, the mean task completion time amounted to M = 159.25 seconds, with SD = 110.41 seconds. The mean number of mouse clicks without Cross Compare was M = 15.92 (SD = 9.48), while it was M = 26.88 (SD = 20.54) with the comparison feature. As multiple hypothesis are being tested, the p-value has been adjusted to counteract the problem of conducting multiple comparisons. By applying the Bonferroni correction ,statistical significance is assumed with $p < 0.0125$. Based on the above means and deviations, the two-tailed p-value for task completion time equals 0.45 and, hence, results in no significant outcome ($p > 0.0125$). For the number of mouse clicks, the p-value is also below the threshold of 0.0125 with $p = 0.02$. Therefore, with the results of the first study, neither of the two null hypotheses can be rejected. While the research focus is the general comparison of using the dashboard with and without Cross Compare, the differences between separate tasks can support discussing the overall findings. Figure 5.1, hence, illustrates each task's completion times and mouse clicks.



(a) Completion time per non-comparison task

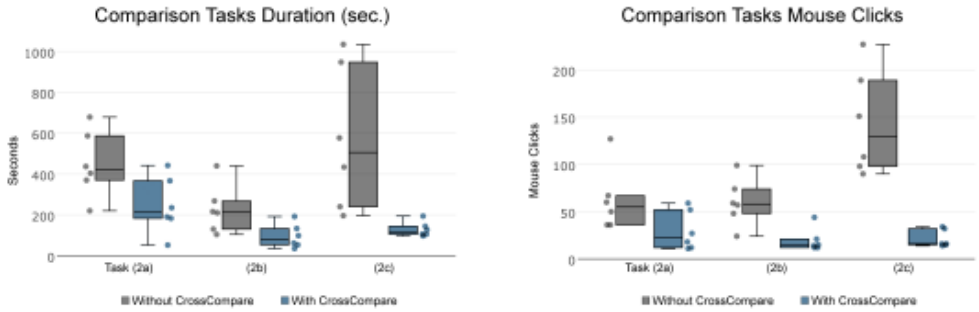


(b) Mouse clicks per non-comparison task

Task with using CrossCompare has the highest average task completion time (M = 234.67s (SD = 143.04s)), while the lowest mean completion time was recorded for Task without the comparison feature (M = 58.00s (SD = 27.66s)). Task (1b) with Cross Compare also required the most mouse clicks on average (M = 35.33 (SD = 21.42)), while Task without the comparison feature required the least (M = 10.67 (SD = 2.94)).

Comparison Tasks Analysis:

As with the first study, each task of the second study–was performed 12 times, once by each participant, in equal proportions either with or without the comparison feature. Each task has, thus, been performed six times with and six times without Cross Compare. Across all tasks of the second study, the task completion time without the comparison feature yielded M = 416.39s (SD = 265.17s). With the comparison feature, the results are M = 156.39s (SD = 107.25s). The mean number of mouse clicks without Cross Compare was M = 88.89 (SD = 54.98), while it was M = 23.56 (SD = 14.78) with the comparison feature. For, both, task completion time and the number of mouse clicks, the p-values are below 0.0005 and, thus, fulfil the requirement for statistical significance of $p < 0.0125$. Hence, for the second study, both null hypotheses can be rejected. Again, the differences between separate tasks will be presented to allow a more thorough discussion of the findings. illustrates each task's completion times and mouse clicks



(a) Completion time per comparison task (b) Mouse clicks per comparison task

Task without using Cross Compare has the highest average task completion time (M = 571.67s (SD = 353.95s)), while the lowest mean completion time was recorded for Task with the comparison feature (M = 95.50s (SD = 58.84s)). Task without CrossC ompare also required the most mouse clicks on average (M = 143.83 (SD = 55.30)), while Task with the comparison feature required the least (M = 19.67 (SD = 12.37)). All mean task completion times, mouse clicks, and task difficulties of the first and second study are summarised, per task, in Appendix E.

6.1. Findings:

User Management (if applicable):

- **Number of active users:** This helps gauge the overall user base and engagement.
- **User growth rate:** Tracks how your user base is expanding over time.
- **User activity patterns:** Identifies which features or content users engage with the most.
- **User segmentation analysis:** Analyze user behavior and preferences based on demographics (age, gender) or other relevant categories.

System Health:

- **Resource utilization:** Monitor CPU, memory, and disk usage to identify potential performance bottlenecks.
- **Uptime and downtime:** Track system uptime and identify periods of downtime for maintenance or troubleshooting.
- **Error rates:** Monitor the frequency of errors and identify potential issues requiring attention.
- **Response times:** Analyze server response times for API requests to ensure smooth user experience.

Content Management (if applicable):

- **Content performance:** Analyze metrics like views, comments, and shares to gauge audience engagement with your content.
- **Popular content:** Identify content pieces that resonate most with your audience.
- **User completion rates:** For forms or interactive content, assess the percentage of users who complete the intended action.
- **Average time spent on content:** Understand how long users typically engage with certain content types.

E-commerce (if applicable):

- **Sales performance:** Track key metrics like total revenue, conversion rate, and average order value.
- **Best-selling products:** Identify which products are generating the most revenue.
- **Customer behavior analysis:** Analyze customer purchase patterns and identify potential upselling or cross-selling opportunities.
- **Inventory management:** Monitor stock levels to prevent stockouts and optimize inventory management.

General Data Analysis:

- **Traffic patterns:** Track website or application traffic to understand user behavior and peak usage periods.

- **Device and OS usage:** Analyze the types of devices and operating systems users are accessing your platform from.
- **Demographic insights:** Gain insights into user demographics (if applicable) to tailor your content or marketing strategies.
- **Conversion funnel analysis:** Identify and address any drop-off points in your conversion funnel (e.g., checkout process).

Remember:

- **Focus on KPIs:** Prioritize displaying the most critical metrics that provide actionable insights for improvement.
- **Comparative Analysis:** Compare data points over time or across different segments to identify trends and patterns.
- **Visualizations:** Utilize appropriate charts and graphs to effectively communicate data insights.
- **Actionable Insights:** Don't just display data; translate it into actionable insights that administrators can use to make informed decisions.

By analyzing the data and understanding these key findings, administrators can gain valuable insights to improve system performance, optimize content strategies, understand user behavior, and ultimately achieve your business goals.

6.2 Advantages and Applications:

6.2.1 Advantages of Admin Dashboard Analysis

Admin dashboard analysis offers a multitude of benefits for businesses and organizations:

- **Improved Decision Making:** By providing real-time and historical data visualizations, dashboards empower administrators to make informed decisions based on concrete evidence rather than intuition or guesswork. Trends, patterns, and anomalies become readily apparent, allowing for data-driven strategies.
- **Enhanced Efficiency:** Dashboards streamline the monitoring process by consolidating key metrics in one central location. This eliminates the need to gather data from multiple sources, saving administrators valuable time and effort.
- **Increased Productivity:** Faster access to insights enables administrators to focus on strategic tasks rather than spending time hunting for data. This leads to a more proactive and productive working style.
- **Improved System Performance:** Real-time monitoring of system health allows for the prompt identification and resolution of potential issues. This prevents downtime, ensures optimal performance, and protects against security risks.

- **Better Resource Allocation:** Data visualization helps administrators understand where resources are being utilized and how efficiently. By analyzing trends, resource allocation can be optimized to better align with the needs and goals of the system.
- **Enhanced User Experience:** By tracking user behavior and engagement, administrators can identify areas for improvement in the user experience. They can then tailor content, features, and functionality to better meet user needs and expectations.
- **Boosted Revenue and Sales (e-commerce):** Analyzing sales performance, customer behavior, and product popularity enables administrators to make data-driven decisions regarding marketing strategies, product offerings, and inventory management. This can lead to increased sales and improved revenue streams.

6.2.2 Applications of Admin Dashboard Analysis

Admin dashboard analysis can be applied to a wide range of scenarios and industries:

- **Website Analytics:** Track website traffic, user behavior, and conversion rates to optimize web content, improve user experience, and drive online marketing strategies.
- **E-commerce Operations:** Monitor sales performance, customer trends, and product analytics to enhance marketing campaigns, personalize customer experiences, and optimize inventory management for better profitability.
- **Social Media Management:** Analyze follower growth, engagement metrics, and content performance to understand audience preferences and tailor social media strategies for maximum impact.
- **Content Management Systems:** Track content viewership, user interaction, and sharing statistics to identify popular content, refine content types, and create targeted content strategies.
- **Help Desk Management:** Monitor ticket volume, resolution times, and customer satisfaction to identify areas for improvement in customer service operations and resource allocation.
- **Server and Network Monitoring:** Real-time monitoring of system health, resource utilization, and performance allows for the proactive identification and resolution of potential issues before they impact uptime or user experience.
- **Human Resources:** Analyze employee performance data, track project progress, and monitor resource assignments to optimize workflows, identify skill gaps, and make informed payroll decisions.

These are just a few examples, and the possibilities for admin dashboard analysis extend to any application or system where data collection and analysis can provide insights for improvement and optimization.

7. Summary, Conclusion And Recommendation

The increasing interest in dashboards as information systems has transformed the business intelligence landscape. BI vendors and service providers are adapting to this ever-developing branch of visual data analytics which has led to a diversification of the BI market towards visual data discovery. These data discovery solutions, however, require licenses, complex deployments, and are intended for company-internal (i.e. performance management) usage. To facilitate dashboard usage as information services outside a company-focused context, additional research efforts are required. Thus, contemporary dashboard usage and design practices have been reviewed to lay the foundation for the development of a Web-based discovery dashboard. An examination of current Web-based dashboard and charting solutions has shown that Web-based dashboard development is feasible and can already provide the user with highly interactive views which allow brushing and linking. However, some discovery features were not available.

Especially the ability to compare various dashboard states, following the principle of focus+context, was yet to be realised. This project, hence, resulted in the development of CrossCompare.js, a JavaScript library to facilitate contextualised comparisons, and its implementation in an exemplary air traffic delay dashboard – according to the requirements of WINGX, who seeks to investigate the marketability of Web-based dashboards. This paper accompanies the development process and outcomes of, both, CrossCompare and the air traffic dashboard. Their following evaluation has shown the benefits of contextualised comparisons and proven the viability of Web-based dashboards. With the release of CrossCompare.js v1.0.0 and the air traffic dashboard, both of which are open-source and built using only freely licensed resources, it is hoped to have advanced and spurred the development of Web-based discovery dashboards. By now, it is fair to assume that Web-based, open-source dashboards are fully feasible for commercial information services and will allow a broader public to take advantage of the manifold benefits of visual analytic

8. Future Research

While a discovery dashboard has been developed, it required substantial development effort and the combination of numerous libraries. The introduction of a full-stack, open-source, Web-based dashboard framework could vastly accelerate the usage and adoption of online dashboards. Further, balancing client- and server-side data processing would be an interesting development project as to optimize performance and allow for – potentially – Big Data analytics. Cross Compare will also undergo further development as to support a wider range of chart types and charting libraries. The popularity and increasing usage of mobile devices also poses challenges to dashboard design, as it renders the common requirement of a single-screen view infeasible. Exploring different variants of the required compromise between screen-size and presenting all information at a glance, might provide valuable insights for mobile dashboard design and dashboard perception in general. Additionally, the differing possibilities for interaction on a mobile device, compared to a desktop machine, could lead to a more intuitive usage experience for all users – regardless of their preferred platform. Based on the experience gathered from the conducted user studies, further experiments could identify and investigate different types of dashboard tasks and – possibly – examine the effect of various dashboard functionalities (such as contextualised comparisons) on the users' satisfaction and effectiveness. A subject of particular interest and, thus, potential topic of future research projects, are the factors that give users confidence in their dashboard usage. Identifying aspects that – besides aesthetics – reinforce the users' confidence might add a new dimension to dashboard development. Finally, it is hoped that this project – and possible future research based on it – contribute to the amendment of the "dearth of research on dashboards.

9. Reference

[1] CGI Group Inc. Next Generation Business Intelligence: Seven steps to improved business intelligence through data discovery, 2011. URL: http://www.cgi.com/sites/ default/files/white-papers/business-intelligence-white-paper.pdf.

[2] Wayne G. Bremser and William P. Wagner. Devoloping dashboards for performance management. The CPA Journal, 83(7):62–67, 2013.

[3] M´aria Bielikov´a, Pavol N´avrat, Daniela Chud´a, Ivan Pol´aˇsek, Michal Barla, Jozef Tvaroˇzek, and Michal Tvaroˇzek. Webification of Software Development: General Outline and the Case of Enterprise Application Development. Global Journal on Technology, Vol 3 (2013): 3rd World Conference on Information Technology (WCIT-2012),03:1157–1162, 2013.

[4] Ogan M. Yigitbasioglu and Oana Velcu. A review of dashboards in performance management: Implications for design and research. International Journal of Accounting Information Systems, 13(1):41–59, March 2012. URL: http://www.sciencedirect. com/science/article/pii/S1467089511000443, doi:10.1016/j.accinf.2011.08. 002.

[5] Solomon Negash. Business intelligence. The Communications of the Association URL:http://aisel.aisnet.org/cgi/viewcontent.cgiarticle=3234&context=cais.

[6] H.P. Luhn. A Business Intelligence System. IBM Journal of Research and Development,2(4):314–319, 1958. doi:10.1147/rd.24.0314.

[7] Seth Grimes. BI at 50 Turns Back to the Future, 2008. URL: http://www.informationweek.com/software/information-management/ bi-at-50-turns-back-to-the-future/d/d-id/1073576?page_number=1.

[8] Thomas H. Davenport. Analytics 3.0. Harcard Business Review, pages 64–72, December 2013.

[9] Wayne W. Eckerson. Performance Dashboards: Measuring, Monitoring, and ManagingYour Business. John Wiley & Sons, 2010.

[10] Lisa Pappas and Lisa Whitman. Riding the technology wave: Effective dashboard datavisualization. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6771 LNCS(PART 1):249–258, 2011. arXiv:9780201398298, doi:10.1007/978-3-642-21793-7_29.41

[11] Stephen Few. Information Dashboard Design - The Effective Visual Communicationof Data. OReilly Media, 2006.

[12] K. Pauwels, T. Ambler, B. H. Clark, P. LaPointe, D. Reibstein, B. Skiera, B. Wierenga, and T. Wiesel. Dashboards as a Service: Why, What, How, and What Research Is Needed? Journal of Service Research, 12(2):175–189, 2009. doi:10.1177/1094670509344213.

[13] Richard A Becker, William S Cleveland, and Murray Hill. Brushing Scatterplots. Technometrisc, 29(2):127–142, 1987. doi:10.2307/1269768.

[14] Daniel Keim and Matthew Ward. Visual Data Mining Techniques. Techniques, 2002. doi:10.1097/ALN.0b013e31825dd7ac.

[15] H Hauser. Interactive Visual Analysis-an Opportunity for Industrial Simulation. SimVis, pages 1–6, 2006. URL: http://www.vrvis.at/publications/pdfs/PB-VRVis-2006-014.pdf.

[16] Johannes Kehrer, Peter Filzmoser, and Helwig Hauser. Brushing moments in interactive visual analysis. Computer Graphics Forum, 29(3):813–822, 2010. doi: 10.1111/j.1467-8659.2009.01697.x.

[17] Rita L. Sallam, Bill Hostmann, Kurt Schlegel, Joao Tapadinhas, Josh Parenteau, and Thomas W. Oestreich. Magic Quadrant for Business Intelligence and Analytics Platforms, 2015. URL: http://www.gartner.com/technology/reprints.do?id=

1-2ACLP1P&ct=150220&st=sb.

[18] Thomas H. Davenport. The Rise of Data Discovery, 2008.

[19] Wan Maseri, Wan Mohamad, Embong Abdullah, and Mohamad Zain Jasni. ImproveKnowledge Visualization through an Interactive Graph-based Dashboard System withKey Performance Indicator : A Case Study of University Dashboard for Higher. pages1–7, 2007.

[20] Doug Henschen. How To Choose 'Advanced' Data Visualization Tools, 2012. URL: http://www.informationweek.com/software/information-management/how-to-choose-advanced-data-visualization-tools/d/d-id/1105480?

[21] B Y Wayne Eckerson. Visual Discovery Tools : Market Segmentation and Product Positioning. (March):1–30, 2013.

[22] Andrew Foley. QlikView Vs. Tableau: Software Showdown, 2015. URL: https://www.clearpointstrategy.com/qlikview-vs-tableau/.

[23] Q-Success. Usage Statistics of JavaScript for Websites, August 2015. URL: http://w3techs.com/technologies/details/cp-javascript/all/all.

[24] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D 3 : Data-Driven Documents.IEEE Transactions on Visualization and Computer Graphics, 17(12):2301–2309, 2011.

[25] Drew Skau. Why D3.js is So Great for Data Visualization, 2013. URL: http://blog.visual.ly/why-d3-js-is-so-great-for-data-visualization/.42

[26] Christophe Viau. Whats behind our Business Infographics Designer? D3.js of course., 2012. URL: http://www.datameer.com/blog/uncategorized/whats-behind-our-business-infographics-designer-d3-js-of-course-2.html.

[27] Scott Murray. Interactive Data Visualization for the Web. O'Reilly Media, Inc., 2013.URL: http://chimera.labs.oreilly.com/books/1230000000345.

[28] M Dewar. Getting Started with D3. 2012.

[29] Nick Qi Zhu. Data Visualization with D3.JS Cookbook. Packt Publishing Ltd., 2013.

[30] Square. Crossfilter - Fast Multidimensional Filtering for Coordinated Views, 2012. URL: http://square.github.io/crossfilter/.

[31] Chris Weaver. Multidimensional visual analysis using cross-filtered views. VAST'08- IEEE Symposium on Visual Analytics Science and Technology, Proceedings, pages163–170, 2008. doi:10.1109/VAST.2008.4677370.

[32] Grant Stanley. Dashboards : Take a closer look at your data. URL: http: //canworksmart.com/three-types-of-dashboards/.

[33] Juice Inc. A Guide to Creating Dashboards People Love to Use, 2009. URL: http: //www.cpoc.org/assets/Data/guide_to_dashboard_design1.pdf.

[34] Boon Wan Tan and Tak Wah Lo. The impact of interface customization on the effect of cognitive style on information system success. Behaviour & Information Technology,10(4):297–310, 1991. doi:10.1080/01449299108924291.

[35] Cody W. Hanson. Chapter 2: Mobile Devices in 2011. Library Technology Reports,47(2):11–23, 2011.

[36] John Deacon. Model-view-controller (MVC) Architecture, 2009. URL: https: //techsimplified2.com/Uploads/Agendas/October28,2011.pdf.

[37] Nick Heidke, Joline Morrison, Mike Morrison, and Eau Claire. Assessing the Effectiveness of the Model View Controller Architecture for Creating Web Applications Science.

[38] W3C. HTML5 is a W3C Recommendation, October 2014. URL: http://www.w3.

org/blog/news/archives/4167.

[39] Julien Sobrier. The move to plugin-free browsers, February 2013. URL: http://research.zscaler.com/2013/02/the-move-to-plugin-free-browsers.html.

[40] Rohan Pearce. The rise and rise of JavaScript, January 2014. URL: http://www.techworld.com.au/article/536950/rise_rise_javascript/.

[41] QlikTech. Qlikview Development and Deployment Architecture, February 2011. URL: http://www.qlikview.com/us//~/media/Files/resource-library/global-us/direct/datasheets/DS-Technical-Brief-Dev-and-Deploy-EN.ashx.

[42] Marc Rueter and Ellie Fields. Tableau for the Enterprise : An Overview for IT,May 2012.43.

[43] Ganesh Iyer. Big Data visualizations using crossfilter and dc.js, 2014. URL: http://ganeshiyer.net/blog/2014/07/19/big-data-visualization-using-crossfilter-and-dc-js/.

[44] Darshit Shah. SmartFilter Not in-memory server-side Crossfilter for BigData, 2015.URL:http://axiomnext.com/blog/smartfilter-not-inmemory-server-side-crossfilter-for-bigdata/.

[45] Stephen Few. Pervasive Hurdles to Effective Dashboard Design. Perceptual Edge, (January):7, 2007.

[46] Patrick Moore Fitz and Chad. Gestalt Theory and Instructional Design, 1993.

[47] Edward R. Tufte. Beautiful Evidence. Graphics Press LLC, 2006.

[48] Qiao Ma. The Effectiveness of Requirements Prioritization Techniques for a Medium toLarge Number of Requirements : A Systematic Literature Review. (November):1–83,2009.

[49] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar,Abdullah Gani, and Samee Ullah Khan. The rise of Big Data on cloud computing:Review and open research issues. Information Systems, 47:98–115, 2014. doi:10.1016/j.is.2014.07.006.

[50] Frederic Lardinois. Move Over 1024768: The Most Popular Screen Resolution On The Web Is Now 1366768, April 2012. URL: http://techcrunch.com/2012/04/11/move-over-1024x768-the-most-popular-screen-resolution-on-the-web-is-now-1366x768/.

[51] Eric W. Weisstein. Great Circle. URL: http://mathworld.wolfram.com/GreatCircle.html.

[52] Eduardo Antonio Cecilio Fernandes. Scrum explained, January 2015. URL: http://www.codeproject.com/Articles/704720/SCRUM-explained.

[53] Jeff Williams, Jannon Frank, Michael Mathews, and Tim Schaub. @use JSDoc, August2015. URL: http://usejsdoc.org/.

[54] Ashley J S Mills. JavaDoc. Technical report, The University Of Birmingham, Birmingham, 2005.

[55] American Statistical Association. Data expo ’09 - Get the data, 2009. URL: http://stat-computing.org/dataexpo/2009/the-data.html.

[56] John Brooke. SUS-A quick and dirty usability scale. Usability evaluation in industry,189:194, 1996.

[57] Jeff Sauro. Measuring Usability with the System Usability Scale (SUS), January 2011.
URL: http://www.measuringu.com/sus.php.

[58] James V Bradley. Complete Counterbalancing of Immediate Sequential Effects in a Latin Square Design. Journal of the American Statistical Association, 53(282):525528,
1958. doi:10.1080/01621459.1958.10501456.

[59] Olive Jean Dunn. Multiple Comparisons Among Means. Journal of the American Statistical Association, 56(293):52–64, 1961. doi:10.2307/2282330.