

Making the Right Choices in Dashboard Design and Development

Spandana Kottur*

*(Department of Computer Science and Engineering, Rashtreeya Vidyalaya College of Engineering, Bengaluru Email: kotturspandana@gmail.com)

Abstract:

A dashboard is a tool used for information management and business intelligence. It provides the business useful insights to make important decisions through a visual representation of information. The dashboard design is important to ensure that the relevant information is conveyed to the user with the right representation, one must also make a choice of technologies and it is important to identify based on requirements of the dashboard and its scale. This paper explores the choice of front-end technologies between AngularJS and ReactJS and back-end technologies like Node.js and Java listing out their various features and their suitability for a web based dashboard.

Keywords —Dashboard, Dashboard Design, AngularJS, ReactJS, Node.js, Java.

I. INTRODUCTION

Business Analytics or Business Intelligence (BI) is becoming essential for top management of any organization to visualize, analyze and prepare the strategic planning for the future. However, the availability of business intelligence tools, the support on the existing infrastructure, usability, scalability, financial commitment and other parameters make it challenging for the Information technology [4]. A dashboard is a tool used for information management and business intelligence. It presents a centralized, interactive means of measuring, monitoring, analyzing, and extracting information that facilitate business decisions. They communicate metrics in an interactive, intuitive, and visual manner [9]. When displayed in the appropriate manner this information can help identify trends, extract information from related metrics which otherwise may not be very obvious. The design of a dashboard would vary based on its functions and the intended users, since major enterprises use dashboards to make big business decisions they use executive dashboards. An executive dashboard gathers and displays

information that top-level stakeholders need to run a company, business, or organization. Executive dashboards function a lot like business dashboards, but the information they provide cater specifically to the needs and expectations of executives. Executives can only afford to gather and understand a limited amount of information at a given time, which means they need instantaneous access to the most relevant information. A dashboard can be utilized to its maximum capacity only if it depicts the right metrics and therefore choosing the most appropriate KPI's is at the core of building an efficient dashboard. KPIs should be measurable, tangible metrics that let each member of the organization and every stakeholder understand how their performance influences the success of the organization.

Advantages

There are several key advantages that make a dashboard a valuable tool for business intelligence.

1. Fast and effective decision-making -: It gives executives, managers and analysts easy and instantaneous access to key performance metrics, that can help monitor performance and processes

for a better and deeper understanding of the business.

2. On demand information: Dashboards clearly convey business goals across the institution and allow users to work towards achieving those goals. This allows everyone to be aware of the goals. Having a customized design allows users to only see the data that is most relevant to them, and they can discard information that is not important.

3. Easily identifiable patterns: Users can immediately see any uneconomical results or negative trends to take quick measures, boosted potential, drill down on charts, reports to further gain complete information, study the current data closely to identify deeper issues and to rectify negative trends.[10]

II. DASHBOARD DESIGN

The purpose of a dashboard is to convey as much information as possible through visual representations. This implies that a dashboard must be carefully designed so as to display maximum possible amount of important information. Although there is no specified set of rules to design a dashboard there are a set of guidelines that must be kept in mind during the design phase. Dashboard design is not just another visualization job. It must be displayed on a single screen and thus it should make optimum use of the space and avoid scrollbars or multiple windows. A static display is preferred, allowing one to focus on the information being represented. The position of the information on the screen is also very important. The top, left and centre of the screen should contain the most crucial information. Dashboards must have drill-down abilities, the depth of which must be agreed among the users during the development of the dashboards. All the layers of the drill down should lead to the main page of the dashboard in a single application. Dashboards must have the capability to display older data. A dashboard that can predict future trends allows the user to make better use of the dashboard. To accommodate predictability, the dashboard must also have the capability for statistical analysis. Displaying expert knowledge on

dashboard ensures that any future risks can be mitigated by providing timely diagnosis to any upcoming issues.

Keeping the visual representation simple and using well known standard notations is crucial. Using red, yellow, and green colours to display the estimation of features is preferred, along with globally accepted representation scales. Simple graphs with individual trends and tables are a suggested way of displaying data. The use of blinkers, flashers, and alarms, are mostly discouraged as they are often times distracting.[7] Discussions conducted in dashboard design reviews found several interesting insights about the way the effectiveness of the information conveyed through a certain visual representation of the data. Data represented as a table is superior to the same data shown in a graph. But graphs are better representation of sales forecast data. Graphs also reduce the information overload when compared to tabular information. [11] The functional and visual needs of a traditional business dashboard can be addressed keeping in mind some ideas while designing frameworks. Three dimensions were identified to focus on in BI dashboard design. An information model should be created to cater to users with various roles performing a variety of tasks to provide data literacy at a number of levels. This framework was extended to form dashboards that are linked to each other through multiple layers supporting the three types of organizational decision making, depending on the type of display, the information to be conveyed and analytical function. These frameworks have some elements in common but are much less complete. Another framework also describes dashboard design as finding balance between 4 dimensions: purposes, users, design features and outcomes. Purposes comprise of monitoring, analysis, planning and communication. Functional features involve, view reconfiguration, notifications, tools used for analysis and drill-down/roll-up capabilities. Functional fit can be used to describe how much the dashboard's function help in achieving the goal. The functional and the visual features are equally

important considering the variety of users involved.[6]These frameworks portray the newer features of the business needs while designing dashboards. An increasing variation in users in terms of tasks and roles, dynamic flexibility in purpose and function, and more option to interact with data for details, are the most crucial things to be kept in mind while designing effective business dashboards

III. FRONT END LANGUAGES

A. Single Page Application

A good dashboard application has displays the most recent data, this means the data being displayed on the page is being constantly refreshed, traditional websites rerender each time a portion of the page is changed which is not very efficient which is why dashboards can be used effectively if they are built as a single page application.

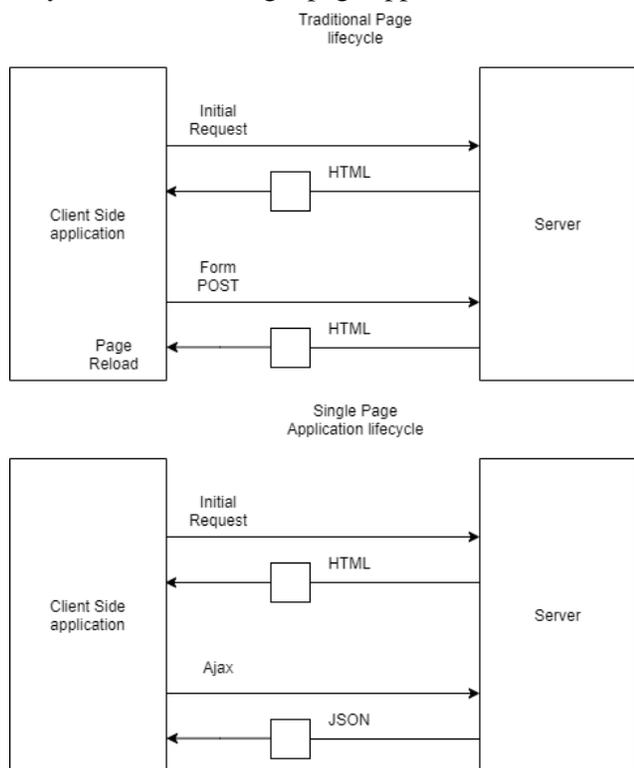


Fig. 1 A Lifecycle of a traditional page vs. life cycle of an SPA

Single Page Application (SPA) consists of independent components that can be modified or updated on their own, without refreshing all the contents of the page so that the whole page is not rerendered every time the user interacts with the application which saves bandwidth and prevents the loading of external files each time. The purpose ensures that subsequent page loads are faster than the traditional request-response cycle. [8]

B. Building a Single Page Application

For large scale web applications built for industry use, developing using a web stack is considered an efficient method to build applications that complete the requirements of the business. JavaScript along with JSX and ES6 are the essential components that enable integration of many other technologies. Javascript is a great way for developers to implement application logic, design front end experience, while also allowing access to the database. Two of the most popular options for developing single page web applications are the MEAN(MongoDb Expressjs Reactjs Nodejs) stack and the MERN(MongoDb Expressjs Angularjs Nodejs) stack. The difference between the two stacks is only with respect to using react or angular to design the user interface. It is important to understand the abilities of both to be able to make the right choice. AngularJS is a front end JavaScript framework whereas ReactJS is a JavaScript library. A framework provides a foundation for writing code. It follows a fixed methodology for the organization of the code. Angular follows the Model-View-Controller architecture.. Both angular and react are used to design a user interface for the presentation layer for an application. React is considerably a better choice for as it shows better performance. AngularJS allows components to listen to data and updating their values at the same time, this is known as two way data binding,each time a change is made to a component a digest cycle is triggered for synchronization of data amongst the layers. This happens to be expensive when a large number of items need to be rendered dynamically. A unidirectional data flow detects any

changes that happen to the state, which works well with bigger datasets where thousands of records are to be created and updated. While state control has improved with Angular 2, React uses simpler methods to handle these state changes.[5]

C. ReactJS

ReactJS is JavaScript library that allows developers to create reusable user interface components. The official React document defines it as follows:

“React is a library for building modular user interfaces. React makes it easier to develop large scale applications that need constant change of data without a large number of page refreshes. In an MVC architecture, it can be used to construct the view . React allows upation of components through state changes thus abstracting the Document Object Model (DOM), thus offering a simple yet high strong application building experience, generally react uses Node.js on the backend. React implements unidirectional data flow that is less prone to errors, providing better control over data[12]

D. AngularJS

AngularJS is an open-source web application framework. Its official documentation defines it as :

AngularJS is a structural framework for dynamic web applications. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application components clearly and succinctly. Its data binding and dependency injection eliminate much of the code to be written, Also it all happens inside the browser, making it a great choice for most server technologies. [13]

AngularJS allows developers to write frontend applications conveniently using the Model-View-Controller architecture.

E. Choosing Between React and Angular

A major difference in the architecture comes from the fact that Angular is a framework while React is just a library. Since React is a library the developer has more control over the structure and

organization of the code whereas in Angular one is forced to follow the MVC architecture. As mentioned earlier Angular also has a lot of inbuilt functions therefore the necessity of using third party libraries decreases making it easier to maintain the code but in case of react using a large number of third party libraries may make it a little difficult to maintain the code. Both React and Angular have things in common however there are some features that make React a better choice for industry level applications such as enterprise dashboards,

- Data can be stored securely
- Latency is less
- The actual code is hidden from the user
- Uses the concept of virtual DOM making resulting in pages rendering faster

In terms of use of backend resources Angular outperforms react as it most of the work happens on the client machine reducing significant load on the servers.[5]Table 1 summarises the major differences in React And Angular, while react is a good choice Angular may be used in cases where the server side costs need to be reduced.

TABLE I
ANGULAR VERSUS REACT

Attributes	Angular	React
Author	Google	Facebook
Language	HTML/Javascript	JSX
Architecture	MVC	View Only
Third party library support	Low	High
DOM	Normal	Virtual
Binding	Two way	Unidirectional
Rendering	Client-side	Server-side

IV. BACKEND LANGUAGES

Architecture of an application is responsible for how quickly dashboards responds to the user’s actions and the amount of load the application handles. In case of dashboards, it also defines the ease with which the user can view various level of information through navigation. It contributes to the overall ease of maintaining and changing the dashboard wit change in requirements. It was previously mentioned that two popular development stacks are MEAN and MERN they both use node.js

as their back end another alternative is to use Java, in the following section the features of both shall be discussed

A. Node.js

Node.js is a server-side platform built on Google Chrome's JavaScript Engine.

Official documentation defines it as follows:

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, convenient for applications that are data intensive and need to run real time across distributed devices. Node.js also provides a large library of variety JavaScript modules which largely simplify the development of web applications. [1]

Node.js is also known for having a large number of supporting libraries with various functions that can reduce the amount of time and effort required to build an application.

B. Java

Java is a cross-platform object-oriented programming language that was released by Sun Microsystems in the year 1995

Java is a general-purpose, class-based, object-oriented programming language designed for having fewer implementation dependencies. It is a computing platform for application development. Java is fast, secure, and reliable. [3]

C. Comparing Node.js and Java

Java is a very popular language and has quite a few languages. Java is widely preferred for its write once run anywhere ability, which means no matter where it was created it can be run on any platform. The java virtual machine allows this kind of platform independence. Node.js on the other hand needs to rebuild modules each time it is run on a different operating system. Java offers better support in terms of IDE's, each IDE is a complete ecosystem in itself which makes writing code easier, Node.js does not have an equally wide variety of IDEs that support it when compared to java. Java

also has a large number of libraries simply because it was around for longer, these libraries are also open source, supported and developed by Java developers, while Node.js modules can be considered as libraries and their modular structures makes it convenient for use, Java libraries offer better quality. Java allows for multi threading where several tasks can be performed simultaneously which makes it a great choice for computation intensive tasks. Node.js is single threaded the tasks are placed one after the other in a queue where they are processed one by one. But Node.js has its own advantages. It was the first tool to allow use of Javascript on the backend. Using javascript as both back end and front-end creates a smooth infrastructure. Java offers only back end development. Node.js also happens to be much easier to learn, it also results in shorter code which makes time taken to create an application much lesser. While Node.js lacks in terms of vertical scalability as it is only single threaded it makes up significantly in terms of horizontal scalability. Both Node.js and Java and Node.js match each other in terms of speed.[2]Table 2 summarizes the features of Node.js and Java.

TABLE III
NODE.JS VERSUS JAVA

Attributes	Java	Node.js
Language	Object oriented Language	Framework using Java script
Threads	Multithreading	Single thread
IDE Support	Multiple Options	Not many options
Third party library support	High	Moderate
Scaling	Good vertical scaling	Good horizontal scaling
Usability	Only Backend	Javascript in both frontend and backend

V. CONCLUSIONS

The paper talks about a dashboard as a business intelligence tool and its role in making executive decisions in an organization. A dashboard must be designed keeping its users in mind it must be static and covered on a single page/screen, it must avoid distracting functionalities. For efficient usage

dashboard must have preferably have statistical analysis capabilities. While there is no right way to express the information, several guidelines are to be kept in mind while designing the dashboard most importantly appropriate charts must be used to display the information to help identify trends that may not be otherwise be obvious. These applications are best built as a single page application as opposed to the traditional application because they can reduce the load of the request-response cycle. To build the visual interface two of the most popular choices ReactJS and AngularJS are discussed While both are good options React outperforms Angular for large scale projects in terms of speed due to the presence of virtual DOM , it also has better third party library support which might be necessary to allow a variety of charts in a given dashboard. Two backend technologies are also reviewed Node.js and Java. , Java is seen as a better choice for industry level applications that are compute intensive as it uses multithreading whereas Node.js is better in terms of scalability and provides a better infrastructure allowing both the front and back end to be in Javascript. It is seen that neither one is better than the other however it depends on the type of application being created.

REFERENCES

- [1] "Node.js Tutorial - Tutorialspoint," Tutorialspoint.com. [Online]. Available: <https://www.tutorialspoint.com/nodejs/index.htm>. [Accessed: 10-May-2021].
- [2] "Node.js vs java: Choosing a back-end language for large app development," Da-14.com. [Online]. Available: <https://da-14.com/blog/nodejs-vs-java-backend-language-large-software-development>. [Accessed: 10-May-2021].
- [3] "Java Tutorial," Tutorialspoint.com. [Online]. Available: <https://www.tutorialspoint.com/java/index.htm>. [Accessed: 10-May-2021].
- [4] Gowthami, K., and MR Pavan Kumar. "Study on business intelligence tools for enterprise dashboard development." *International Research Journal of Engineering and Technology (IRJET)* 4, no. 04 (2017): 2987-2992.
- [5] Aggarwal, Sanchit, and Jyoti Verma. "Comparative analysis of MEAN stack and MERN stack." *International Journal of Recent Research Aspects* 5.1 (2018): 127-32.
- [6] Sarikaya, Alper, et al. "What do we talk about when we talk about dashboards?." *IEEE transactions on visualization and computer graphics* 25.1 (2018): 682-692.
- [7] Bugwandeem, Keshika, and Marius Ungerer. "Exploring the design of performance dashboards in relation to achieving organisational strategic goals." *South African Journal of Industrial Engineering* 30.2 (2019): 161-175.
- [8] Jadhav, Madhuri A., Balkrishna R. Sawant, and Anushree Deshmukh. "Single page application using angularjs." *International Journal of Computer Science and Information Technologies* 6.3 (2015): 2876-2879.
- [9] S. Durcevic, "What is A data dashboard? Definition, meaning & examples," Datapine.com, 29-Jul-2020. [Online]. Available: <https://www.datapine.com/blog/data-dashboards-definition-examples-templates/>. [Accessed: 10-May-2021].
- [10] "Dashboards and dashboarding," Logianalytics.com, 29-Sep-2015. [Online]. Available: <https://www.logianalytics.com/resources/bi-encyclopedia/dashboards-dashboarding/>. [Accessed: 10-May-2021].
- [11] Tokola, Henri, et al. "Designing manufacturing dashboards on the basis of a Key Performance Indicator survey." *Procedia CIRP* 57 (2016): 619-624.
- [12] Aggarwal, Sanchit. "Modern web-development using reactjs." *International Journal of Recent Research Aspects* 5.1 (2018): 2349-7688.
- [13] "AngularJS - Overview," Tutorialspoint.com. [Online]. Available: https://www.tutorialspoint.com/angularjs/angularjs_overview.htm. [Accessed: 30-May-2021]