ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 9, Issue 2, pp: (18-23), Month: October 2022 – March 2023, Available at: www.paperpublications.org

Emerging Technologies involvement in E-learning: An expository using Progressive Web Application

[1]Emmanuel Adediran, [2]Wumi Ajayi (Ph. D)

DOI: https://doi.org/10.5281/zenodo.7442150

Published Date: 15-December-2022

Abstract: **School Management System using Progressive Web Application is aimed at Network Independent for Student Management system which allows the students to check through their details without necessary using data logging-in via their matriculation number, since the developed System will not allow matriculation numbers outside Lead City University. The main goal of this thesis is to develop a Progressive Web Application School Management System and provide an interface that will be used for all the necessary details of School Management. The objectives in achieving the set goal are: Designing of the database, designing of the design of the progressive web application school management system and the development of the whole system with HTML, CSS, PHP, JavaScript, React, Xamp Server. The system is targeted to helping students take their studies in a more advanced fashion breaking away from the conventional way of using papers and manual processing of doing their registration all the time. The School Management System using Progressive Web Application makes use of features such as service workers, push notification, indexedDB, Background sync is deployed for the implementation of the system. The system will give a more developed and user-friendly access as well as it will take less time to complete its execution. The cost of implementing the system is minimal compared to the cost of other development of other School Management System. The system is deployed into the Department of Computer Science in Lead City University.**

Keywords: **School Management System, Emerging Technologies, Progressive Web Application.**

1. INTRODUCTION

Because of its productivity and suitability, the application of data innovation in educational administration has exploded[1]. PCs are seen to have the potential to make a significant contribution to classroom educating, learning, and organizing. Equipment, programming, systems administration, and staff development that have gone into bringing data and communication technology (ICT) into schools will be seen as beneficial if there is confirmation that it has an equal impact on school execution and adequacy[2]. Because of its efficacy and suitability, the application of data innovation in educational administration has exploded. School administrators who previously expended a great deal of time and effort dealing with perplexing assignment difficulties (e.g., staffing, asset allocation, and timetabling) and monitoring school duties now have more options thanks to enhanced innovation[3]. With the decentralization of work projects and their coordination in an intelligent correspondence structure, data advances work.

Fast technological alterations are influencing how goods and administrations are planned all the time. The turn of events rehearses for apps have been developed since the introduction of cell phones. Because engineers must write, test, and maintain programs for each of the numerous phases and devices, traditional portable application development has been described as a time-consuming and costly process[4]. As a result, most adaptable programming engineers are employing a variety of strategies to improve the progression of applications from local to half-breed programs[5]. Furthermore, a few ways for cross-stage development have been given out as an optional path for long-term local progress.

The portable web is based on a web application that is typically cross-platform. This type of software adheres to industry-standard languages like HTML5, CSS3, and JavaScript, which provide full app portability between platforms like Android and Apple[6]. The arrival of cross-stage development frameworks like Cordova, PhoneGap, and Ionic has made the construction of applications for many platforms a lot easier[7]. It also reduces the cost and effort of marketing and allows for a faster time to market. Crossplatform techniques, on the other hand, don't always succeed because of their dependency on a WebView (an in-app included program portion)[8] and program constraints, which results in substandard execution and client experience.

Site pages today are more than just static HTML markup. We've progressed from a Web without JavaScript, where all connections should have been handled by the server, to using JavaScript just for activities and minor convenience features, to developing entire applications in JavaScript that mostly communicate with servers behind the scenes. When JavaScript gained the ability to execute HTTP requests and retrieve or save data in the foundation, a technology known as AJAX at the time, it was only used for minor purposes. Single Page Applications (SPAs) were developed later, with the application being layered once at the start and then transmitted in JSON or XML later. This presented additional hurdles to JavaScript developers, such as delivering layouts in a timely manner. This presented new hurdles to JavaScript developers, such as supplying layouts in the program, monitoring application state, and steering (the planning between application state and the URL showed in the location bar of a program). In addition to Web development, cell phones and associated applications, often known as local applications nowadays, were developed (instead of web applications). These programs used the full capabilities of the devices they were running on and could accomplish a lot more than a web application in the beginning. This meant that a company that wanted to reach as many customers as possible had to produce many versions of the same app: for the web, the workplace, and a few other platforms. Later efforts like Cordova and Electron made it possible to use JavaScript and other web developments in programs as well as local mobile and desktop applications, but the goal of making the web the most widely used application platform remained.

Several additional JavaScript augmentations were produced under the Web Platform project, interfaces that give JavaScript access to various features of the device on which the program runs. The Location API, which allows a site to access a device's GPS location, and the Screen Capture API, which allows a site to capture the screen of a program or another application, are two examples.

2. RELATED WORKS

Abstract, Introduction, Related Works, Method, Results Conclusion, Reference

Mobile Native to Hybrid

Designing for mobile devices initially and gradually transitioning to larger devices is known as the "mobile-first" approach to design and content. It is a reaction to the expanding need for better mobile experiences. A major factor in the rise of mobile-first design was Google's announcement at the 2010 Mobile World Congress that they would be adopting the strategy and urged other designers to follow suit: "I think it's now the collaborative mission of all of us to make mobile the answer to pretty much everything"[7].

Native Applications

Both the use of mobile devices and native applications has grown. Applications created specifically for a platform or device are known as native applications. They have certain definite advantages because they are created on a particular hardware and operating system[8].

Users receive the quickest, most dependable, most responsive experience from native applications. They can make advantage of hardware and software designed specifically for the smartphone, like the camera, microphone, and push notifications. Because designing for a certain platform is simpler, platform conventions can be incorporated into the UI/UX[9].

It is not surprising that native applications are replacing time spent in browsers because of their exceptionally quick loading times and offline accessibility, to mention a couple of reasons. It was further established in development of mobile applications in 2017, and as can be seen, mobile app usage has increased by 6% yearly[10]. Particularly when buying, users spend more time on apps and less time on browsers. Diverse research shows that as consumers continue to transfer their purchasing to e-commerce through mobile shopping apps, the amount of time spent in e-commerce applications has significantly increased (54%) as can be observed[11].

Nevertheless, the reach of mobile websites is almost three times greater than that of native apps. Most users aren't paying attention and only utilize their top three apps for 80% of their time. The typical user installs zero native programs in a month, indicating that users generally dislike installing them[12]. One explanation could be because they require frequent user updates and take up storage space on their devices.

Other drawbacks to native programs exist as well. They require distinct development for every operating system, each of which has its own codebase, and are more difficult to grow. They typically require a team with these kinds of skill sets and take longer to develop than online or hybrid applications.

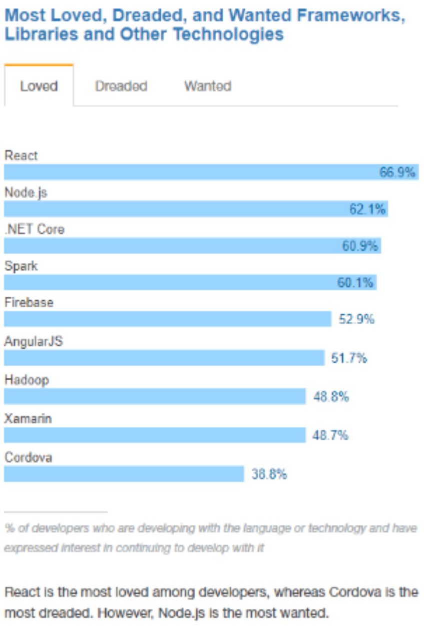
Hybrid Applications

Applications that are hybrid operate on various platforms and behave like native apps. They are created utilizing a mix of native and web technologies. In a native application container, they are technically web applications. A mobile device can execute applications that have access to the device's functions, such as the camera and GPS. They are released through a native application store, much like native programs. Examples of application stores that assess these programs are Apple and Google. The most well-known hybrid applications right now include, to mention a couple, Netflix and Amazon. These applications can be created in a variety of methods, including using the React Native or Apache Cordova frameworks[13].

The cost-effectiveness and speed of delivery of hybrid applications built with frameworks are two distinct advantages. The UX experience of an application created using this kind of framework is still inferior to that of native apps and has various disadvantages, such as slowness and phantom clicking while scrolling, lack of offline functionality, and the potential for the application to execute slowly depending on its complexity[14]. Because, for instance, the Apache Cordova framework does not by default allow fingerprint scanning, capabilities may also be somewhat constrained[15].

3. METHOD

React is a JavaScript library, used for building UI. Facebook first introduced React in the year 2013. It was made open source two years later. React uses the virtual DOM for manipulating the UI as a result it gives blazing fast performance and making site most interactive as shown in figure 7[16.] Designing simple views for each state in application, React smoothly updates and renders just the needed components when the data changes. Declarative views in React make code easy to predict and debug.[17] React uses component-based structure. Component logic are written in JavaScript instead of templates which makes the easy passage of rich data through the web app and keeping state out of the DOM. Problems are solved by creating reusable components and when components get complex, they are broken down into smaller and simpler ones. React components are similar to JavaScript function.[18]



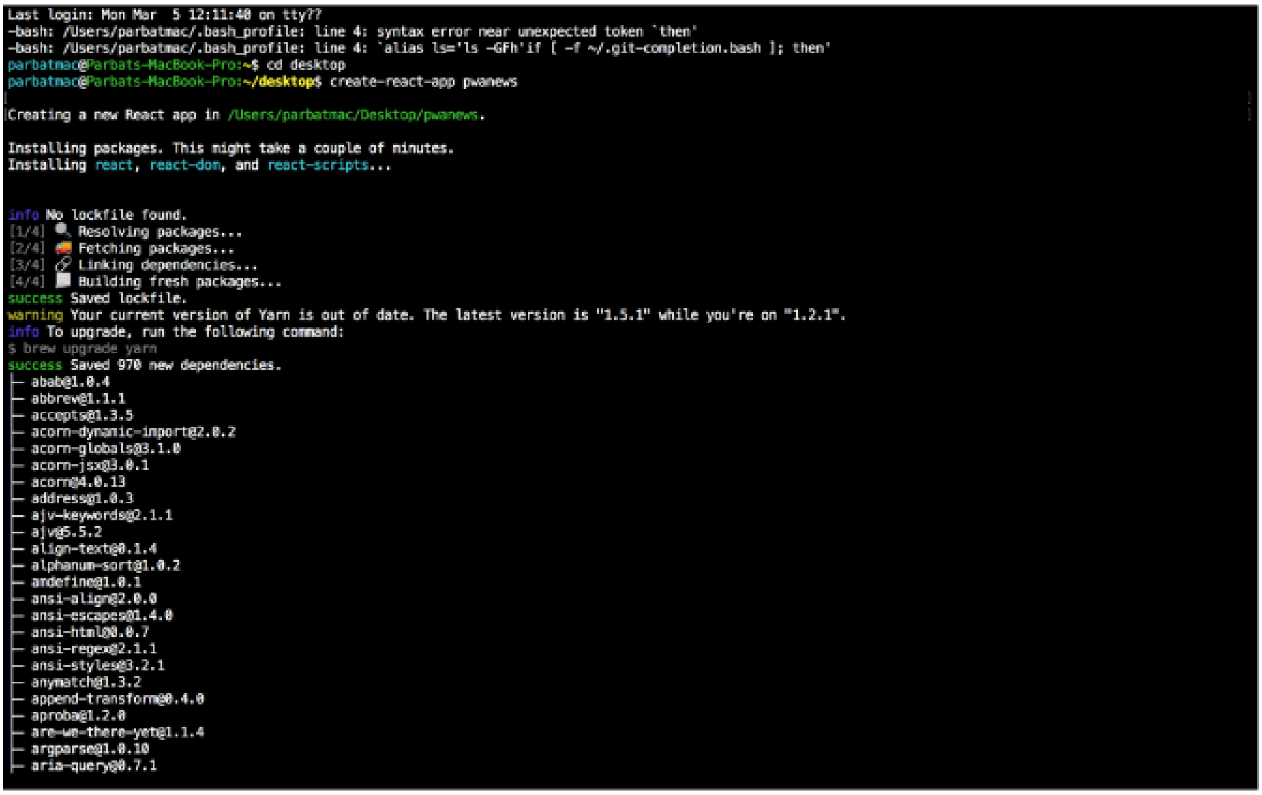
% of developers who are developing with the language or technology and have expressed interest in continuing to develop with it

React is the most loved among developers, whereas Cordova is the most dreaded. However, Node.js is the most wanted.

Figure 1

As shown in Figure 3.2, React was the most loved technology in 2017 at 66.9% whereas the least loved one is Cordova with 38.8% only. React uses an XML-like syntax JSX, a syntax extension to the ECMAScript to describe the component's DOM representation. It looks similar to that of the mark-up language HTML or XML but it is JavaScript centric. It makes code more readable and writing JSX gives the feel of writing HTML. It is based on separation of concerns rather than technology as it combines mark-up, style and behaviour of components in one file rather than having each separate files. However, it is not compulsory to use JSX in React application but the use of JSX helps for easy development, error handling and increase performance of React application.[19]

Create react app is a tool to get started with React.js app development. Create react app with pre-configured Webpack, Bable and other necessary tools makes an app development process smooth, and hassle free. It also comes pre-configured with Service Worker, which is one of the key components of PWA. Create react app can be easily installed from the terminal with the command: npm install -g create-react-app, as shown in Figure below.



```
Last login: Mon Mar  5 12:11:48 on tty??
-bash: /Users/parbatmac/.bash_profile: line 4: syntax error near unexpected token `then'
-bash: /Users/parbatmac/.bash_profile: line 4: `alias ls='ls -GFh'if [ -f ~/.git-completion.bash ]; then'
parbatmac@Parbats-MacBook-Pro:~$ cd desktop
parbatmac@Parbats-MacBook-Pro:~/desktop$ create-react-app pwanews

Creating a new React app in /Users/parbatmac/Desktop/pwanews.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts...


info No lockfile found.
[1/4] 🔍 Resolving packages...
[2/4] 🚚 Fetching packages...
[3/4] 🔗 Linking dependencies...
[4/4] 📃 Building fresh packages...
success Saved lockfile.
warning Your current version of Yarn is out of date. The latest version is "1.5.1" while you're on "1.2.1".
info To upgrade, run the following command:
$ brew upgrade yarn
success Saved 970 new dependencies.
├─ abab@1.0.4
├─ abbrev@1.1.1
├─ accepts@1.3.5
├─ acorn-dynamic-import@2.0.2
├─ acorn-globals@3.1.0
├─ acorn-jsx@3.0.1
├─ acorn@4.0.13
├─ address@1.0.3
├─ ajv-keywords@2.1.1
├─ ajv@5.5.2
├─ align-text@0.1.4
├─ alphanum-sort@1.0.2
├─ amdefine@1.0.1
├─ ansi-align@2.0.0
├─ ansi-escapes@1.4.0
├─ ansi-html@0.0.7
├─ ansi-regex@2.1.1
├─ ansi-styles@3.2.1
├─ anymatch@1.3.2
├─ append-transform@0.4.0
├─ aproba@1.2.0
├─ are-we-there-yet@1.1.4
├─ argparse@1.0.10
├─ aria-query@0.7.1
```

Figure 2: Installing create-react-app project (Researcher, Adediran E. 2022)

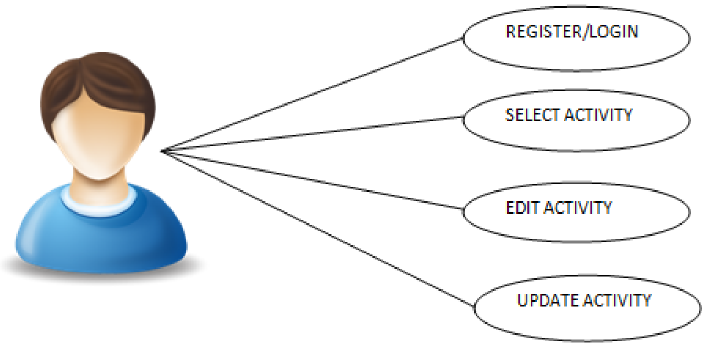
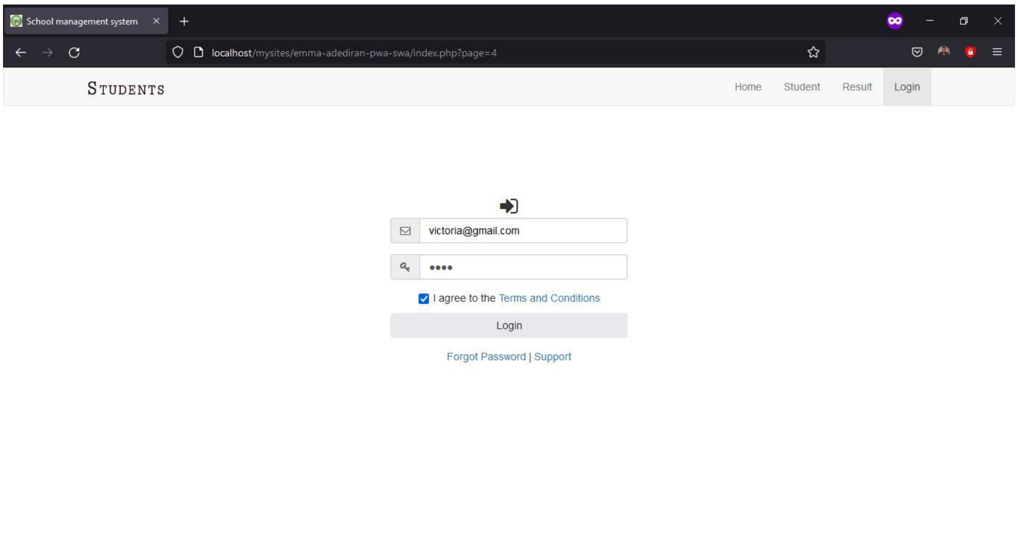


Figure 3: Use Case diagram

4. RESULTS



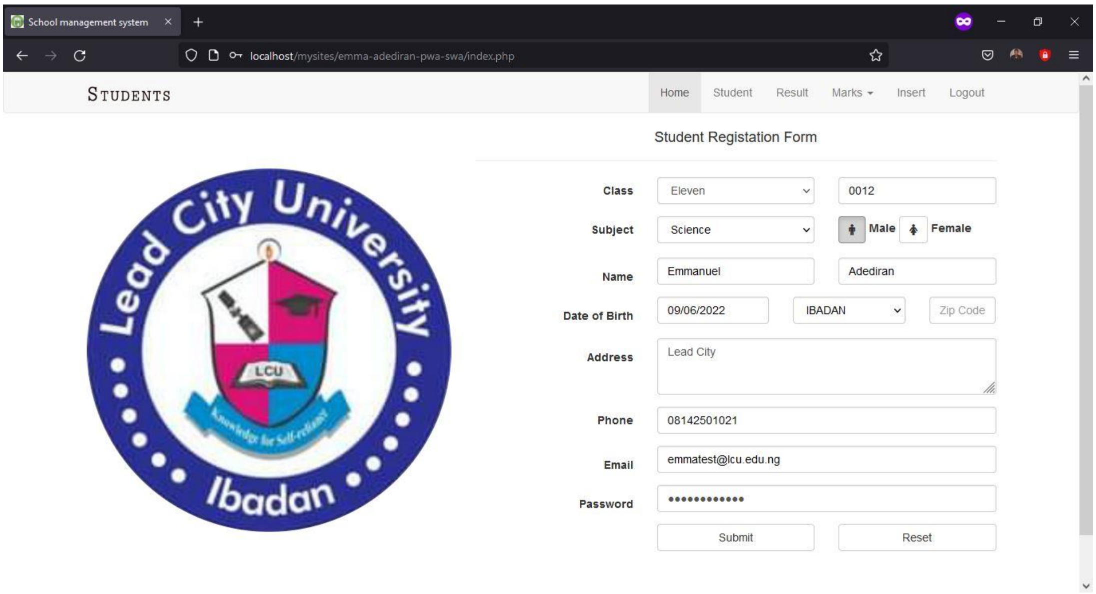
localhost/mysites/emma-adediran-pwa-swa/index.php?page=4

victoria@gmail.com

••••

☑ I agree to the Terms and Conditions

Figure 4



localhost/mysites/emma-adediran-pwa-swa/index.php

Student Registation Form

Class	Eleven	0012	
Subject	Science	Male	Female
Name	Emmanuel	Adediran	
Date of Birth	09/06/2022	IBADAN	Zip Code
Address	Lead City		
Phone	08142501021		
Email	emmatest@lcu.edu.ng		
Password	••••••••••••		

Figure 5

5. CONCLUSION

In conclusion, Instructive organizations can basically keep their understudy records by utilizing understudy the board frameworks. The manual framework makes it hard to achieve this objective on the grounds that the data is scattered, frequently excess, and social event relevant information could take a ton of time. This undertaking settle every one of these issues. This framework supports safeguarding the association's understudy data set. The supervisor may promptly get to it and keep it secure for quite a while without making any changes.

In this research explore the problem of Network Independence was solved. In view of this, techniques and approaches to make applications function offline have been introduced. A combination of technologies is used to accomplish this: Service Workers to manage page requests (for instance, by storing them offline), the Cache API to store network request responses offline (which is very helpful for storing site assets), and client-side data storage technologies like Web Storage and IndexedDB to store application data offline. Modern web apps can be created to offer a great user experience to browsers with all the necessary capabilities, and a passable (although less flashy) experience to browsers with less capability which was done in this project. Best practices like progressive enhancement was used. PWAs are cross-browser due to the progressive improvement they use. Researchers should therefore consider how various browser implementations of some PWA features and technologies differ from one another.

REFERENCES

[1] Shah, Madiha.. *Impact of Management Information Systems (MIS) on School Administration: What the Literature Says.* Procedia - Social and Behavioral Sciences 116, no. February (February): 2014, 2799–2804.

[2] Seagraves, Liz, and Summer Kenesson. *The Impact of ICT in Schools – a Landscape Review Professor Rae Condie and Bob Munro*. 2006 http://webarchive.nationalarchives.gov.uk/20101102103654/publications.becta.org.uk/download.cfm?resID=28221.

[3] Castells, Manuel, *The Rise of the Network Society*. Oxford, Uk ; Cambridge, Mass., Usa: 1996 B. Blackwell.

[4] A. Biørn-Hansen, T.A. Majchrzak, and T.M. Grønli, April. *Progressive web apps: The possible web-native unifier for mobile development.* In International Conference on Web Information Systems and Technologies 2017 (Vol. 2, pp. 344-351). SCITEPRESS.

[5] Malavolta, *Beyond native apps: web technologies to the rescue! (keynote)*. In Proceedings of the 1st International Workshop on Mobile Development 2016 (pp. 1-2).

[6] Malavolta, G. Procaccianti, P. Noorland, and P. Vukmirovic, *Assessing the impact of service workers on the energy efficiency of progressive web apps.* In 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft) 2017 May (pp. 35-45). IEEE.

[7] T. A Majchrzak, A. Biørn-Hansen, and T.M. Grønli, *Progressive web apps: the definite approach to Cross-Platform development* 2018

[8] W. Jobe, W. *Native Apps vs. Mobile Web Apps*. International Journal of Interactive Mobile Technologies, 2013, *7*(4).

[9] H. Joo, *A study on understanding of UI and UX, and understanding of design according to user interface change. International Journal of Applied Engineering Research, 12(20),* 2017 9931-9935.

[10] M. Satyanarayanan, G. Klas, M. Silva, & S. Mangiante, *The seminal role of edge-native applications.* In 2019 IEEE International Conference on Edge Computing (EDGE) 2019, July (pp. 33-40). IEEE.

[11] Wang, Z. Feng, S. George, R. Iyengar, P. Pillai, & M. Satyanarayanan, *Towards scalable edge-native applications*. In Proceedings of the 4th ACM/IEEE Symposium on Edge Computing 2019, November (pp. 152-165).

[12] N. Kratzke, & P. C Quint. *Understanding cloud-native applications after 10 years of cloud computing-a systematic mapping study.* Journal of Systems and Software, 2017 *126*, 1-16.

[13] C. Griffith, *Mobile App Development with Ionic, Revised Edition: Cross-Platform Apps with Ionic, Angular, and Cordova*. "O'Reilly Media, Inc." 2017.

[14] G. Toffetti, S. Brunner, M. Blöchlinger, J. Spillner, & T.M Bohnert, *Self-managing cloud-native applications: Design, implementation, and experience.* Future Generation Computer Systems, 72, 2017 165-179.

[15] P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, & K. Tzoumas, *Apache flink: Stream and batch processing in a single engine.* Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 36(4), 2015.