Genie Enterprise Resource Planning for Small Medium Enterprises Implementing Single Page Web Application

J Gunawan and R R Kosala*
Computer Science Department, Faculty of Computing and Media, Bina Nusantara University, Jakarta, Indonesia 11480

*Corresponding author: rkosala@binus.edu

Abstract. The exponential growth of Small Medium Enterprises (SMEs) in Indonesia has led to the greater competition among them. However, their business processes are often inefficiently managed, which increase the cost of operation. The current Enterprise Resource Planning system could address the problem, but the price is relatively expensive for current condition of SME in Indonesia. The problem continues when the business operation run in different places, where data inconsistencies reduce operation efficiency. Cloud-based ERP is a possible solution, but it is relatively slow to fully maximize the benefit. Therefore, new affordable solution with remote access, which use less internet bandwidth is needed to answer problems in implementing ERP. This study aims to analyze the problem and use the findings to provide a prototype that could address the existing problem. Through implementation of various technologies such as Single Page Application, Gzip Compression, and Webpack Module Bundler, the author successfully developed the prototype of ERP system. Several tests, including compatibility test and user acceptance test were performed to evaluate the prototype. Although this prototype has not been able cover the whole modules, the test results showed a satisfactory result in terms of accessibility and basic functionality.

Keywords: SME; ERP; Web Application; Single Page Application (SPA)

1. Introduction

According to data published by Ministry of Co-operative and SMEs Republic of Indonesia, Small Medium Enterprise (SME) almost covers 99.9% of the total business in Indonesia. SMEs had contributed 60.34% of Indonesian Gross Domestic Product (GDP) and 97.22% of employment opportunities [1]. However, considering the tight competition that will continuously arise, SMEs should have an added value which enables them to minimize cost and maximize profits. Improving the effectiveness and efficiency of daily operational business processes is one option. Unfortunately, majority of SMEs in Indonesia is still operating their business manually where data is not being managed properly with a defined operational procedure [2]. One of supporting technology which could address this problem is Enterprise Resource Planning system.

Enterprise Resource Planning (ERP) is a software to simplify and automate most of the business processes within an integrated system [3]. By implementing ERP, SME could achieve better productivity by providing better customer services, reducing lead time, and providing faster information [3]. However, only a small number of the SMEs implements ERP due to its relatively
expensive implementation cost. The author expects to propose a suitable solution that could address the existing challenges and meet SMEs’ requirements by incorporating current web technology. The study will focus on developing the client side of the system along with an attempt to improve the application performance in terms of minimizing data transfer; particularly to respond the needs of SMEs in field of retail and wholesale industry in Jakarta. Several additional steps such as testing and pilot implementation at an SME, remote access to the application from various devices, and research work on technology comparison for saving bandwidth to access cloud system will also be included in this research.

The aim is to deliver a working prototype of ERP which affordable, accessed remotely, and using less bandwidth consumption. Therefore, the cost of implementation could be minimized and overall performance could be enhanced which would be the benefits for SME in terms of improving the efficiency and effectiveness of managing business process.

2. Theoretical Foundation

2.1 General Theories

Cloud computing is a type of Internet-based computing that provides resources and data to computers on request. Through the implementation of cloud computing, resources management would be easier and more configurable [4]. Users are capable of managing their data stored in a distant place. Cloud computing shared the computing resources to achieve economy of scale and coherence of the system.

Web Application is a software application runs in a web browser, where the client interacts with the server to do business logics [5]. There are two major design patterns that usually used for a web application development, the multi-page web applications (MPA) and the single page web applications (SPA). MPA is a traditional web application that needs to request a new page from the server whenever the application needs to display or submit data to the server. On the other hand, SPA is a web application that fits all user interfaces on a single web page to produce a similar user experience as a desktop application [6]. The main purpose of SPA is to eliminate the need of reloading the page during the use of the application. This could be achieved by shifting some parts of server’s responsibilities to the client side, thus the server requirements could be significantly being reduced [7].

Multitier architecture is a client-server architecture that separates the software into three major functions, consist of a presentation, application processing, and data management. Three-tier are the most common multitier architecture which composed of a presentation tier, a domain logic tier, and a data storage tier that are maintained as independent modules [8]. Through its separation of concerns, each layer can be upgraded or replaced independently.

2.2 Development Technologies

The system will be developed as a cloud-based ERP in a form of web application, where the client side use an open source client-side MVC frameworks in Javascript, AngularJS framework. This will enable automatic view updates by extending basic HTML to implement application logic [9]. AngularJS aims to support the development of interactive SPA and incorporates MVVM architecture models to separate presentation, data, and logic components while developing client-side web applications [10]. Two main features are offered: directive which allows an invention of new HTML syntax specific to the application or reusable component, and two-way data binding which allows the view to be automatically updated whenever changed observed in the model, as well as updating model whenever related changed observed in the view. The server side will be developed using NodeJS as an open source Javascript platform environment to develop fast and scalable server-side network applications. NodeJS could handle lots of connection concurrently because of its main feature as an event-driven and non-blocking I/O model [11].

For data storage, the system will be developed using CQRS pattern which combine SQL for write model and NoSQL for read model. For SQL database, the system will use MySQL as the popular open-source relational database management system (RDBMS). As a relational database system, data is
stored in a structured collection in form of table relation and used a unique key to identify each data row [12]. SQL database provides reliability by enforcing strict data integration through an explicit relationship and ACID (Atomicity, Consistency, Isolation, and Durability) properties which is suitable for write model. While for NoSQL database, the system will use MongoDB which uses a document-oriented system [13]. There are two major differences between document-oriented system and RDBMS. The first one is unlike a row in a table, the document has no constrained to a certain schema. Therefore, two documents in a collection could have completely different elements. The other one is MongoDB allows changing in scheme dynamically, enabling adjustment without having to remodel existing database. MongoDB also provides create, read, update, and delete operations without neglecting the performance, availability and/or scalability. This can be achieved because MongoDB has auto-scaling features and an excellent mirroring system thus allowing the system grow as the changing of user requirements over time.

In addition to SPA, this system will also implement compression and bundling method to reduce the amount of data transfer. For compression, the system would implement Gzip compression. Gzip (GNU zip) is a file compression utility which implements the DEFLATE compression algorithm, a combination of LZ77 and Huffman coding [14]. Gzip compression would produce a compressed file with a gzip format. This format is used in HTTP compression to reduce the data size transfer and accelerate the process of transferring HTML and other resources on the World Wide Web. While Webpack Module Bundler is used for bundling application source code in convenient chunks to be loaded from the server. Webpack offers two special features which are loader and code splitting [15]. Loader enables user to bundle various types of modules such as CSS files, images, and JavaScript by generating static assets which represent those modules. While code splitting enables the source code to be divided into an on-demand loaded chunk to maintain a low initial download. A configuration file would be needed to determine which files are going to be served by Webpack.

2.3. Methodology

The research used a combination of first and second-hand data collected through survey and from the internet. The designing and development of proposed solution will be done using agile software development method approach which focuses on delivering working product iteratively [16]. A pilot implementation testing will be conducted to evaluate the prototype.

3. Problem Analysis

3.1 Existing Problem

A survey was conducted to understand how SMEs conduct their operational business process. The results showed that half of the respondents do not know about ERP concept. Furthermore, 43.8% respondents still manage their business using traditional manual input on physical documents to record and manipulate their operational data, many of which do not aware of the existence of application software which could help them. Some even perceived an application software is an unaffordable solution for their business. By using the manual recording system, 57.1% of the respondents complained about the length of time required and data consistency.

Despite the awareness regarding ERP concept, all respondents who use application software do not implement ERP software in their business operation. They rely on Microsoft Excel as a common solution. However, Microsoft Excel could not replace features offered by ERP in general such as generating and analyzing financial data automatically. Furthermore, the survey highlights main concerns of respondents in using ERP software. The software’s simplicity and ease of use is the concern of 71.9% respondents. Then, half of the respondents wish to get a complete feature with an affordable price. Other concerns are related to supporting remote access and preserving data consistency. From an interview with one SME already using an application software, the lack of data integration is also a concern. The company uses a separate inter-module software to manage inventory and financial data which makes it highly effortful to achieve data consistency between each module,
especially when the company establishes several branches. Other problems are the cost, and the loading time related to speed performance. Based on the survey results, the respondents considered the range of 100,000–150,000 rupiah monthly as a suitable and affordable price for an ERP software. The longer the time required for loading and performing several tasks would reduce the effectiveness of business processes and slows down the productivity level. Furthermore, performance and low costs offered by the cloud-based system would be sacrificed without proper internet speed and sufficient bandwidth to support.

3.2 Proposed Solution

Based on the analysis in the previous section, the main problem is the lack of appropriate ERP design to suit the SMEs in Jakarta, specifically in field of retail and wholesale industry. The proposed solution aim to satisfy the SMEs requirements using cloud-based system, and will focus on delivering essential basic features relevant for retail and wholesale business such as inventory management, sales, and purchase transaction, along with financial reports. This will be coupled with features enabling remote access while maintaining data consistency through an integrated system. To make the price affordable, the author would reduce data size transfer and light usage of resources in the server by implementing supported technology, such as Single Page Application, Compression and Bundling.

As the actualization, the author will develop an ERP software, namely “GENIE”. GENIE would be implemented as a cloud-based ERP with two main building blocks, which are nodeJS for server side and AngularJS for client side. The author’s focus on the development of GENIE is the implementation of SPA model that requires a client to download all part of the web during the initial request and remain as a client side application which able to perform several tasks and logic independently. The communication that occurs with the server will only include data communication in the form of JSON which will be accessed through REST API. Compared to the MPA approach which loads all resources for each page request, the SPA only loads the required data. Therefore, it will reduce the amount of traffic between client and server.

4. Design and Implementation

4.1 System Architecture

The proposed solution of GENIE will be implemented with a top-level system architecture. As a cloud based system, users as clients could access the system in the form of web application through a web browser using various devices. While, cloud server would process each of client’s requests and return the corresponding response. As a client-server architecture, GENIE will implement the three-tier architecture which separate frontend, backend, and data storage as independent modules. Data storage will be handled using MySQL and mongoDB to combine SQL and NoSQL capabilities. Backend as a server side application will be developed using nodeJS and accessed through REST API. While, frontend as a client side application will be developed using Single Page Application (SPA) model and Model-View-ViewModel (MVVM) architecture through AngularJS framework. The architecture is as shown in Figure 1.

Figure 1. Top Level Architecture
The architecture of AngularJS framework would separate user interface, data definition, and logic. This architecture allows the separation of concerns which creates an organized and maintainable code. The following are the essential components in this architecture.

- **View**—provide user interface by displaying the content of HTML page with additional CSS style. It is also responsible for receiving user interaction which able to change the data in Model directly via two-way data binding or call corresponding function in the controller.
- **Controller**—contains logic in the form of JavaScript which able to modify data in Model or perform certain action.
- **Service**—has a similar function as controller, but has a main function to interact with the server and perform basic CRUD operation on the data in Model.
- **Model**—represents data which would be used to update view and send data to the controller via two-way data binding.

### 4.2 Modules

As the initial target, GENIE will focus the essential basic modules that fit SMEs requirements such as buying and selling transaction, inventory management, and financial report; which are mainly the scope of software accounting in general. Therefore, GENIE will cover four main modules which are User, Inventory, Sales and Purchase, and Finance module as illustrated in Figure 2.

![Offered Modules in GENIE](image)

**Figure 2.** Offered Modules in GENIE

### 4.3 Use Case

The use case diagram describes the general use case of the system which includes specific modules. User module focus on user management and access permission to the system. Inventory Module serves as inventory management to manage all records including stock record and detailed information of each product stored in the system. Sales and Purchase Module enables users to record and manage all sales and purchase transaction data. Finance Module provides features to generate financial reports and manage accounting data. The use case diagram is shown in Figure 3.

![Use Case Diagram of GENIE System](image)

**Figure 3.** Use Case Diagram of GENIE System
CRUD Module is the basic functionality offered in the system. Manage objects is the base use case in CRUD Module where object is a term that could represent an entity in the system, such as User, Role, Branch, etc. Then, it would be divided into four basic use cases in accordance with its specific function of CRUD along with specified permission for each of them, as shown in Figure 4.

4.4 Application of Technology
4.4.1 Single Page Application
Evolving from the traditional web application which implement Multi Page Application (MPA), Single Page Application (SPA) presents to enhance performance and user experience. SPA propose a concept of creating a rich client-side application with more capabilities to reduce server workload and data transfer by shifting some part of server’s responsibilities to the client side. Additional business logics are added to the client side to achieve this purpose. Those responsibilities include handling user interaction, performing navigation, and managing how data is presented in the view.

With those capabilities in the client side, SPA attempts to fit all user interfaces in a single web page. During the initial load, all resources including HTML page, JavaScript, and CSS will be downloaded from server to build the initial view. This is called Hard Navigation, which uses the same concept as the traditional web navigation. After successfully building the initial view, SPA allows user to perform in-page navigation or Soft Navigation by rendering new pages with existing resources. By implementing both types of navigation, SPA able to eliminate the need of reloading and requesting new pages while navigating throughout the website. The remaining resource that should be requested to the server is the content of the new page. Therefore, SPA only performs data exchange process with the server for any subsequent requests after the initial load. Data would be transferred in the form of JSON format and accessed through REST API. Through this lightweight communication, SPA able to reduce the amount of data transfer to be requested from the server.

As the implementation, AngularJS framework will be used as a tool for creating single page web application. For the application structure design, each entity in the system will be handled on a separate module to create more organized and maintainable codes. Each module consists of several JavaScript and HTML files which specifically used to manage certain entity. The distribution of files along with their specific functions within a module could be seen in the following Figure 5.
• Service.js—provides interfaces of REST API to communicate with server.
• Page.html—provides user interface for the main page to show the list of objects for a certain entity.
• Page Controller.js—provides logic to manage and handle user interaction occurs in the main page.
• Modal.html—provides user interface for the modal window to manipulate single object with the corresponding form.
• Modal Controller.js—provides logic to manage and handle user interaction occurs in the modal window.

4.4.2 Gzip Compression
Compression is an effective way to reduce file size without losing the essential function of its original purposes. Gzip compression enables web applications to perform HTTP compression to reduce the amount of data transfer between client and server. Applying Gzip compression involves two signs that will be sent and received by the client and server respectively. The client needs to tell the server that it accepts compressed content by including Accept- Encoding: gzip, deflate in the request header. If the server has been set to perform the compression, it will give a response to the client with Content-Encoding: gzip included in the header to indicate that the response contains a compressed content. Then, the client need to decompress the response before proceeding another request.

4.4.3 Webpack Module Bundler
Module bundle is used as a tool for bundling several resource files into one or more static asset to represent the whole resource. Webpack capable to bundle various types of modules including JavaScript, CSS, and image by using corresponding loader. Webpack configuration file needs to be specified to determine which files are going to be bundled. Bundling will improve the efficiency of the communication process between clients and server. Instead of sending several files separately, server only needs to send one bundled code with several modules included. Therefore, the number of files that should be requested to the server will be reduced which also contributes to the efforts of reducing data transfer. These three technologies share a common goal to reduce server workloads and the amount of data transferred between client and server., which in turn will enable GENIE to provide a solution with faster loading time and bandwidth-efficient.

4.5 Technology Test
To support statements in the previous section, the author performs further technology test to investigate the results of applying the technology to sample files. This test covers the use of each technology. Chrome Developer Tools is used as the testing tools to inspect several aspects during the test.

4.5.1 Single Page Application
This test aims to compare the performance of SPA with MPA in terms of the total number of requests while navigating from one page to another page. For a proper comparison, the author will compare GENIE application in the form of both implementations. Then, the number of total requests while performing the same navigation will be inspected and compared. The results are shown in Table 1.

| Summary of Comparison between MPA and SPA Test | Total Request while Navigating |
|-----------------------------------------------|-------------------------------|
| Multi Page Application (MPA)                  | 20 requests                   |
| Single Page Application (SPA)                  | 2 requests                    |

The implementation of SPA in GENIE has been proved to have a less number of the total requests while navigating from User page to Role page. This reduction occurs because of the SPA able to
eliminate the need of reloading several resources which already being loaded on the first page. This result indicates that the implementation of SPA could improve the performance of regular MPA with its contribution to the reduction of the amount of data transfer required during the navigation process.

4.5.2 Gzip Compression
In gzip compression test, sample files will be divided into three categories in terms of file size to assess the effectiveness of its application in each category. Then, the author will compare the file size and load time of each sample file before and after applying gzip compression.

After comparing the test results, the use of gzip compression has been proved to reduce sample files in both aspects including file size and load time. In terms of file size, compression showed a significant result for smaller sized file which managed to get up to 72.7% reduction. However, the opposite result is found in load time aspect where the most effective time reduction occurs in a larger sized file. It is because the compression involves several stages including the decompression process which make it less effective for smaller sized file. The results are shown in Table 2.

| Summary of Compression Test | File Size (Data Transfer) | Load Time |
|-----------------------------|--------------------------|-----------|
|                             | without compression | with compression | reduced  |
| Small File Size (< 50 KB)   | 36.3 KB                  | 9.9 KB      | 72.7 %   |
| Medium File Size (50-100 KB)| 91 KB                    | 32.4 KB     | 64.4 %   |
| Large File Size (>150 KB)   | 157 KB                   | 55.5 KB     | 64.6 %   |

4.5.3 Webpack Module Bundler
In webpack module bundler test, sample files will cover all customized JavaScript files in GENIE resource files. Three aspects will be inspected for each request before and after applying webpack module bundler, including total request, load time and the amount of data transfer. Table 3 displayed the result of the bundling test.

| Summary of Bundling Test | Total Request | Data Transfer | Load Time |
|--------------------------|---------------|---------------|-----------|
| without bundling         | 54            | 169 KB        | 1.36 s    |
| with bundling            | 1             | 50.5 KB       | 0.6 s     |
| reduced                  | -             | 64.8 %        | 55.9 %    |

Through bundling process, the amount of request will be reduced because the request of one bundle will represent other resource files. In this test, 54 resource files are bundled into one static file which means requesting one bundled file is equivalent to requesting 54 resource files independently. This will accelerate the download time and reduce load time. The bundling process also reduce the amount of data transfer where the bundled file size has been reduced significantly.
4.6 Pilot Testing and User Acceptance Test

For pilot testing, GENIE will be tested on an SME as the target market of this system. The User Acceptance Test is done along with the pilot testing to evaluate user satisfaction of the prototype, and to assess whether the system met user’s requirement. In pilot testing, employees were asked to access the system via genie.id and try out various functions available using a sample data. The employee doing the data inputs is then asked to give feedback and assessment of the tested prototype.

The following are the criteria covered in user acceptance test:

• User convenience in performing basic operations for data manipulation on both master and transaction data.
• Usefulness of the offered functions on each module to simplify specific task.
• System performance in terms of processing speed and loading time.
• Conformity of the offered modules in satisfying user requirements.
• User satisfaction of the offered solution.

In summary, UAT results showed satisfactory result although the current prototype only provides limited features. The user feels convenient in performing basic operations on both master and transaction data, and the modules are considered helpful in simplifying tasks. The users also satisfied with the processing speed and loading time. In general, the system satisfies the company’s needs in managing the operational business process.

5. Discussion

The test results showed a satisfactory result of the current prototype. GENIE has succeeded in reducing the size of data transfers by applying SPA, Gzip compression, and Webpack module bundler. The technology test results showed that SPA reduced the data transfer by eliminating the need of reloading the page while navigating the system. Meanwhile, Gzip enabled the HTTP compression that also reduced the file size transferred between client and server. The use of compression improve the quality of data transmission significantly by more than 50% in size, cutting the load time. Moreover, several resource files could be bundled into one static asset to make transfer more effective using Webpack module bundler. The bundling test showed similar result where data transfer and load time reduced by more than 50%. Evidently, the technologies save the bandwidth on a cloud-based system.

A pilot testing on an SME was conducted to assess the functionality of the prototype. However, this prototype was unable to cover the entire functionality per the initial design due to constraints in the development process. The user acceptance test was also conducted to evaluate general issues and basic functionality of the system, and the user satisfied with the general performance as well as convenience in performing basic functionality. Furthermore, the UAT result also indicates that the offered modules accommodated the essential functions and able to simplify data management processes. Despite the positive results, the user hopes the system provide a more complete functionality for better result.

6. Conclusion

Despite the significant growth of SME in Indonesia, it is unfortunate that these SMEs are still managing their daily operational business manually. The results of conducted survey and interview targeting SMEs in the field of retail and wholesale business in Jakarta indicate that there is a lack of proper knowledge and implementation of ERP as a business solution. Moreover, the existing cloud-based ERP also have several issues such as high-priced offers packages and slow page loading time. Therefore, the author proposed a suitable solution which could address the existing challenges.

This research aims to provide a cloud-based ERP system which is affordable, enable remote access, and consume less bandwidth. Two possible approaches to achieve this are minimizing server work, and maximizing available server resources. The study focused on minimizing server work by reducing the amount of data transfer between server and client. Three technologies were then chosen to be
applied, namely Single Page Application, Gzip compression, and Webpack module bundler. In terms of functionality, the offered modules include managing transaction, inventory, and financial data as the target market focused on SMEs in retail and wholesale business.

As the actualization, a prototype of cloud-based ERP software “GENIE” was developed. GENIE consist of two main building blocks, which implement nodeJS for server side and AngularJS for client side. The author’s scope is focused on developing client side with an AngularJS framework to implement SPA. As a result, a prototype of GENIE has been successfully developed. Several tests were performed to evaluate the prototype. Although this prototype has not been able to cover the whole modules, the test results showed a satisfactory result in terms of accessibility and basic functionality. Furthermore, the implementation of proposed technologies has managed to produce a web application with a minimum data transfer.

Through this system, users able to perform data management and data monitoring via internet access with a minimum amount of data transfer required. It relates to the objective of this project to create a solution for SMEs in operating their daily business process by providing a cloud-based ERP system at an affordable price and minimum bandwidth consumption. Since the system has not been developed completely, it could not be implemented yet. Therefore, further development and several improvements are needed to enhance the system implementation.

References
[1] Ministry of Cooperation and SMEs Republic of Indonesia Development of micro, small, and medium enterprises (smes) and large enterprises. Online: http://www.depkop.go.id/berita-informasi/data-informasi/data-umkm
[2] Handayani P W, Hidayanto A N, and Budi I 2013 Business process requirements for indonesian small medium enterprises (smes) in implementing enterprise resource planning (erp) and erp systems comparison Journal of Computers 8(9) 2437-2441
[3] Saputro J W, Handayani P W, Hidayanto A N, and Budi I 2010 Roadmap of enterprise resource planning (erp) research for small and medium enterprises (smes) in indonesia International Conference on Advanced Computer Science and Information Systems (ICACISIS)
[4] Raihana G F H 2012 Cloud ERP–a solution model International Journal of Computer Science and Information Technology & Security 2(1) 76-79
[5] Nations D 2016 Improve your understanding of web applications Online: https://www.lifewire.com/what-is-a-web-application-3486637
[6] Shimanorvsky S 2015 Multi page web applications vs single page web applications Online: http://www.eikospartners.com/blog/multi-page-web-applications-vs.-single-page-web-applications
[7] Mikowski M S and Powell J C 2013 Single page web applications (B and W)
[8] Eckerson W W 1995 Three tier client/server architectures: achieving scalability, performance, and efficiency in client/server applications Open Information Systems 3(20) 46-50
[9] Darwin P B and Kozlowski P 2013 AngularJS web application development (Packt Publ)
[10] Jain N, Mangal P, and Mehta D 2015 AngularJS: A modern mvc framework in javascript Journal of Global Research in Computer Science 5(12) 17-23
[11] About nodejs Online: https://nodejs.org/en/
[12] A tutorial on database concepts, sql using mysql Online: http://www.atlasindia.com/sql.htm
[13] Nosql databases explained Online: https://www.mongodb.com/nosql-explained
[14] Introduction of gzip Online: http://www.gzip.org/
[15] What is webpack Online: http://webpack.github.io/docs/what-is-webpack.html
[16] Burger R 2016 The ultimate guide to agile software development Online: http://blog.capterra.com/the-ultimate-guide-to-agile-software-development/