

A FRAMEWORK TO ELIMINATE DEVELOPER INVOLVEMENT IN DYNAMIC REPORTING SYSTEM

Imran Md Jelas

Faculty of Computer and Mathematical Sciences,
Universiti Teknologi MARA, Perak Branch, Tapah Campus,
35400 Tapah Road, Perak, Malaysia
Author Correspondence, e-mail: imran499@perak.uitm.edu.my

Received: 20 September 2018 / Accepted: 26 October 2018 / Published online: 15 December 2018

ABSTRACT

General dynamic reporting system is used to display general tabulated data to users. Building a general dynamic reporting system is a tedious and time-consuming process. In this paper, we proposed a framework to eliminate a developer involvement in a dynamic reporting system. We proposed a dynamic reporting system to be developed in three separated modules namely: i. table manipulation module, ii. data manipulation module and iii. reporting module. Finally, in this paper we focus on designing the generic master table for table manipulation module namely *master_table* and *master_column*.

Keywords: Reporting System; Dynamic Reporting System; Generic Reporting System

1. INTRODUCTION

Reporting system is a systematic method to provide and receive data and information [1]. Reporting system can be distinctly classified into static reporting system or dynamic reporting system. The information in static reporting system is implausible to change over time while the information in dynamic reporting system do change over time [2]. The data and information in static reporting system is usually hardcoded whereas data and information in dynamic reporting system will be stored in a database system.

Presently, there are a lot of existing dynamic reporting systems such as those discussed in [3 - 9]. However, all discussed reporting systems are specific for certain field as summarize in Table 1. It is safe to conclude that, a general reporting system is not usually discussed and studied.

Table 1. Summarize of Recent Studied Reporting System

Type of Reporting System	Reference
Banking and Finance	3
Environment Management	4 – 5
Healthcare Management	6 – 7
Network Management	8 – 9

General reporting system is used to display general tabulated data to end user. In Malaysia, general tabulated data may come from various sources such as the Department of Statistics Malaysia [10] and the Institut Darul Ridzuan [11]. They share abundant general tabulated data on national statistics, education, environment et cetera, for Malaysia and Perak respectively. But most of the data set is only shared either as comma-separated values (CSV) or Excel format which means end users must first download the data and later manipulate it before they can view the data. This can be safely concluded as immediate result from general reporting system is unusually discussed and studied.

Developing a reporting system is a time consuming, tedious and endless tasks [12]. Reporting system for general tabulated data are usually built in-house. However, in-house reporting system is prone to change regularly during the system operation.

Therefore, we proposed a dynamic reporting system to ease and produce efficient reporting system for general tabulated data. In this paper, we will present the planning level and analysis level to build a dynamic reporting system for general tabulated data with the aim to eliminate programmer extensive programmer involvement.

2. PLANNING STAGE

Typically, the dynamic reporting system involved three-stage user usage. A programmer is responsible to develop a system that later can be used by the clerk. A clerk is accountable to ensure required data are to make available to the users. Figure 1 summarizes the link between programmer, clerk and user in three-stage user usage.

**Fig.1.** Three-Stage User in Reporting System

Typical dynamic reporting system requires an extensive involvement of a programmer before a clerk and a user can utilize the system. Figure 2 summarizes the typical dynamic reporting system environment.

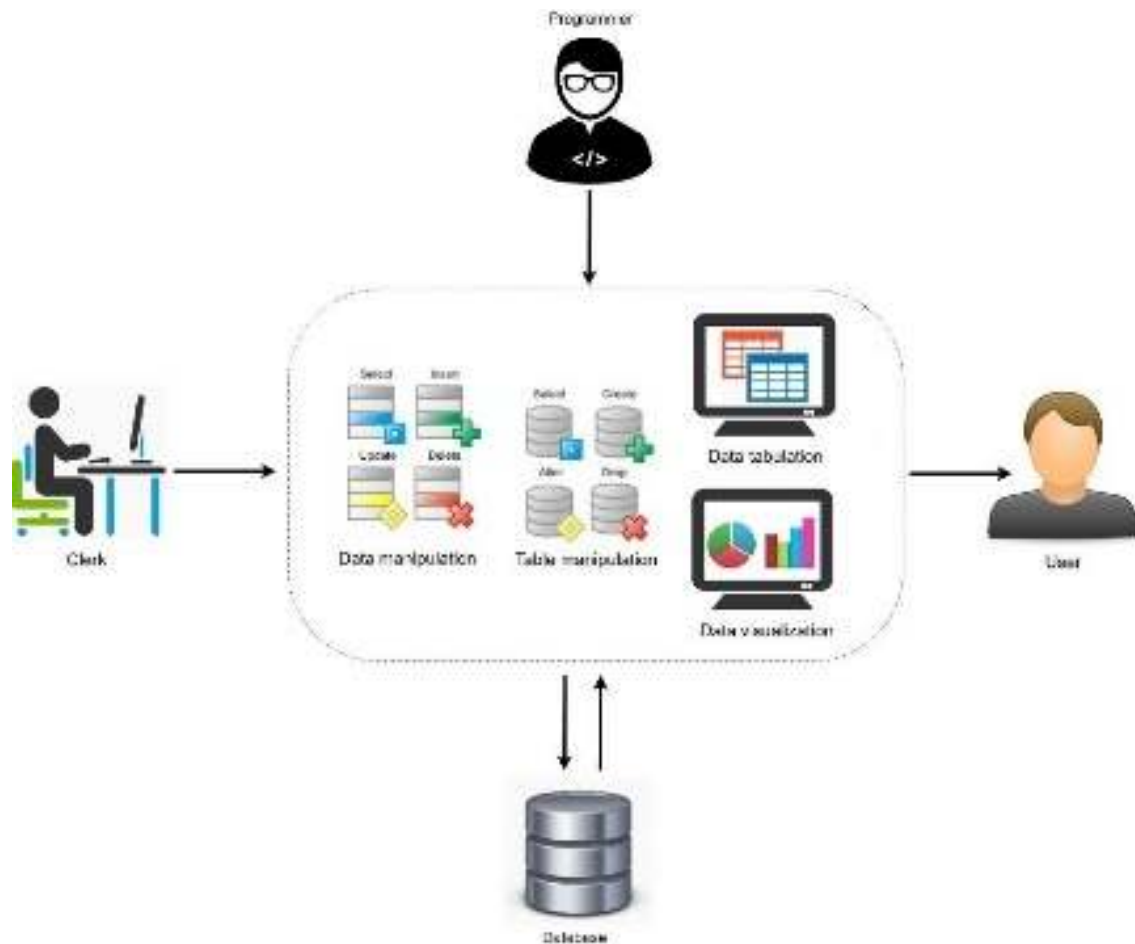


Fig.2. Typical Dynamic Reporting System

First, a programmer must make available table for manipulation. The programmer must ensure all required tables are created including all the entity inside the table itself such as column name and its data type. Second, a programmer must accommodate the clerk with data manipulation module such as view module, insert module, update module and delete module. Third, a programmer must furnish users with reporting module such as data tabulation module and data visualization module. The data tabulation module will display all the data from selected table into a table format and the data visualization module will summarize data into graphical chart format.

3. ANALYSING STAGE

3.1 Proposed Framework for Dynamic Reporting System

The main goal of this paper is to eliminate extensive programmer involvement in dynamic reporting system. However, it is not feasible to simply disregard the programmer's role. The programmer's main task such as table manipulation, data manipulation module and reporting module must still be catered.

Frequently, in any dynamic reporting system, the programmer will prepare the table for manipulation using database administrator module. However, aside from security purposes and the tediousness of database administrator module, it is not feasible to allow the clerk to access database administrator module.

To eliminate extensive programmer involvement in dynamic reporting system we suggest a three separated module: i. table manipulation module, ii. data manipulation module and iii. reporting module.

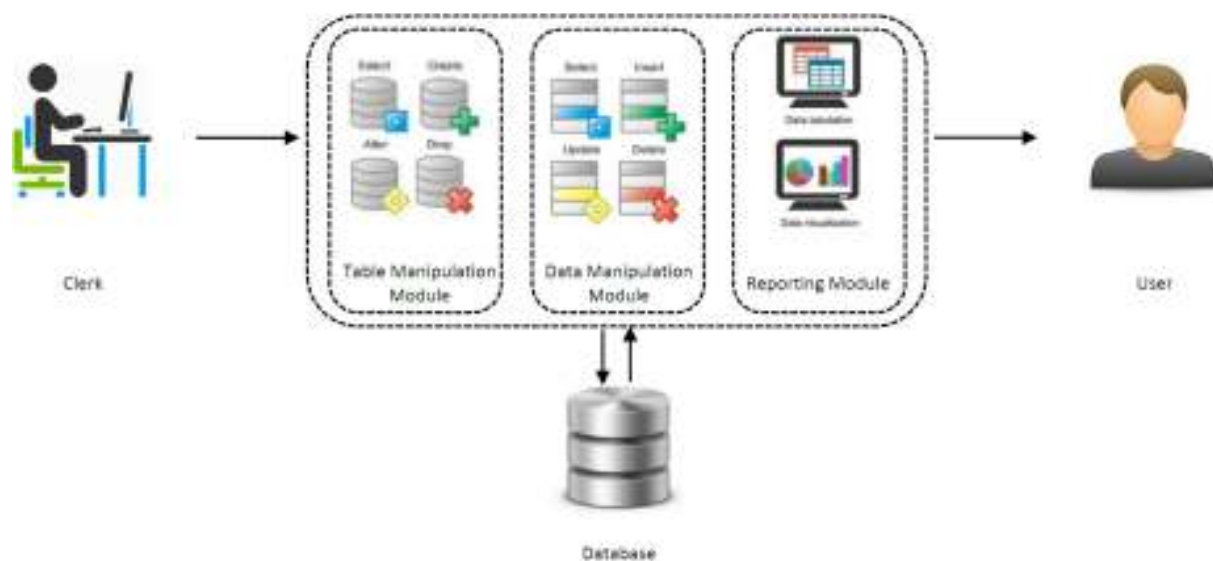


Fig.3. Proposed dynamic reporting system

Table manipulation module will allow the clerk to create a table including all the entity inside the table itself such as column name and its data type. Data manipulation module will allow the clerk to select, insert, update and delete data into the created table. The reporting module will allow users to view data in the data tabulation module and data visualization module. Figure 3 summarizes the proposed dynamic reporting system.

Later in this paper, will focus our discussion on creating a required generic table for the table manipulation module.

3.2 Table Manipulation

We suggest a generic table manipulation module to be provided to the clerk. This will allow the clerk to manage the table and all the entities inside the table itself such as column name and its data type.

Before we can create a table manipulation module, we need to observe the table title and table name. Any input from the clerk will be used as table title and table name. The table title will be used as table data title that later display to clerk and user and hyperlink text throughout the system. Meanwhile, for the table name the same input from clerk will be converted in to small letters with additional three rules: i. symbol will ignore, ii. empty space to be replaced with underscore and iii. limit to only 64 characters. Table name will be used as the actual table name for the system reference and the hyperlink address throughout the system.

Both the table title and table name will later be stored in a table named *master_table* that will be the reference to create a complete table manipulation module later. Table 2 summarizes the requirement for table title and table name in *master_table*.

Table 2. Summarize of *master_table* requirement

<i>master_table</i>	
User Input	Any user input
Title	Maintain User Input
Name	<ul style="list-style-type: none"> • Convert to small letter • Convert space to underscore • Ignore symbol • Limit to only 64 characters

3.3 Column Manipulation

In our analysis, we found out that the general tabulated data type usually comprises of round number, real number or short text. Therefore, other available data types such as date and time, spatial, and long text can be ignored. Apart from that, not all numerical data type is required as we already concluded that tabulated data usually comprise round number and real number. This conclusion lead to a simpler table manipulation module especially on the column data type initialization.

Since column data type initialization has been simplified, handover of the table manipulation from a programmer to clerk is possible. Meanwhile, column title and column name will follow

the same convention as discussed previously for table name and table title respectively. Both column title and column name will later be stored in a table named *master_column* that will also be the reference to create a complete table manipulation module later. However, *master_column* table will be provided with additional column to store column data type. Table 3 summarizes the requirement for column title, column name and column data type in *master_column*.

Table 3. Summarize of *master_column* requirement

<i>master_column</i>	
User Input	Any user input
Title	Maintain User Input
Name	<ul style="list-style-type: none"> • Convert to small letter • Convert space to underscore • Ignore symbol • Limit to only 64 characters
Data type	<ul style="list-style-type: none"> • round number int(11) • real number decimal(10,3) • short text varchar(10)

4. DESIGNING STAGE

4.1 Designing Generic Master Table

Before we can proceed with a complete design of table manipulation module, we need to design the core table used in our system namely *master_table* and *master_column*. This is important since both tables will be used as the main reference in our system.

4.2 Designing *master_table* Table

Previously we have discussed the requirement for *master_table* table that will be the main reference throughout our system. For the *master_table* we need two main columns namely *table_name* and *table_title* respectively. Apart from that, we also need to have a *table_id* as a primary key. Figure 4 shows the query to create *master_table*.

```
CREATE TABLE `master_table`
  (`table_id` INT(11) NOT NULL AUTO_INCREMENT ,
  `table_title` VARCHAR(100) NOT NULL ,
  `table_name` VARCHAR(65) NOT NULL ,
```

```
PRIMARY KEY (`table_id`));
```

Fig.4. Query to create *master_table*.

4.3 Designing *master_column* Table

Earlier we have also discussed the requirement for *master_column* table which will be the main reference throughout our system. For *master_column* we also need to prepare three main columns namely *column_name*, *column_title* and *column_datatype*. Apart from that, we also need to have *column_id* as a primary key and *table_id* as reference key to the *master_table* table. Figure 5 shows the query to create *master_column*.

```
CREATE TABLE `master_column`
( `column_id` INT NOT NULL AUTO_INCREMENT ,
  `column_title` VARCHAR(100) NOT NULL ,
  `column_name` VARCHAR(65) NOT NULL ,
  `column_datatype` VARCHAR(50) NOT NULL ,
  `table_id` INT NOT NULL ,
  PRIMARY KEY (`column_id`));
```

Fig.5. Query to create *master_column*.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a dynamic reporting system with the aims to eliminate extensive programmer involvement in dynamic reporting system. We proposed a dynamic reporting system consisting of the module: i. table manipulation module, ii. data manipulation module and iii. reporting module. Since this paper only focuses on designing table manipulation module, we later narrow our focus on designing a generic master table namely *master_table* and *master_column*.

For future works, we will continue to discuss on developing a complete table manipulation module and interface that will later allow a clerk to create a table including all the entity inside the table itself such as column name and its data type.

6. REFERENCES

- [1] Teamreporter, "Reporting Systems (Overview with tools, software, best practices and alternatives)," Retrieved from <http://www.teamreporterapp.com/reporting-systems>
- [2] Oracle, "Dynamic and Static Reports," Retrieved from https://docs.oracle.com/cd/E12379_01/doc/her.650/admin_ref_web_ui_help/frameset.htm?ch03s01s01.html

- [3] S. Ghosh, "Loan delinquency in banking systems: How effective are credit reporting systems?," *Research in International Business and Finance*, 2018, Retrieved from <https://doi.org/10.1016/j.ribaf.2018.07.011>.
- [4] X. Wang, X. Ding, A. Dou, L. Wang and D. Wang, "Study on GIS and RS based seismic prevention and disaster reduction management information system," *IEEE International Geoscience and Remote Sensing Symposium*, pp 4686 – 4689, 2007.
- [5] H. Jiejun, C. Wei, C. Ting, Z. Yunjun and L. Ye, "Management information system for pollution sources census based on GIS," *2nd IEEE International Conference on Information Management and Engineering*, pp 411 – 414, 2010.
- [6] M. Monawe, M.G. Chawani, G. Kapokosa and C. Moyo, "Strengthening Health Management Information Systems in Malawi: Gaps and opportunities," *IST-Africa Conference*, pp 1 – 7, 2015.
- [7] A.H. Aiken and P.A. Hudgins, "Neck Imaging Reporting and Data System," *Magnetic Resonance Imaging Clinics of North America*, Volume 26, Issue 1, pp 51 – 62, 2018.
- [8] R. Khan and S.U. Khan, "Design and implementation of an automated network monitoring and reporting back system," *Journal of Industrial Information Integration*, Volume 9, pp 24-34, 2018.
- [9] F. Liu and F. Hao, "Web Service Integrated Cloud Computing Management Information System," *7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, Volume 2, pp 288 – 293, 2015.
- [10] Department of Statistics Malaysia, "Open Data," Retrieved from https://www.dosm.gov.my/v1/index.php?r=column3/accordion&menu_id=aHhRYUpWS3B4VXIYaVBOeUF0WFpWUT09
- [11] Institut Darul Ridzuan, "Perak Data," Retrieved from <http://info.idrperak.my/index/home>
- [12] M.S. Desouki, "Dynamic Reports in Administrative Applications," *2nd International Conference on Information & Communication Technologies*, Volume 1, pp 478 – 482, 2006.