A Survey On Knowledgeable Medical Care Explosion System

Dinesh P S [1], Magesh kumar B 2, Dr. G. Nalinipriya [3]

[1] *Assistant Professor, Bannari Amman Institute of Technology, Erode, Tamil nadu, India*
[2]*Assistant Professor, Bannari Amman Institute of Technology, Erode, Tamil nadu, India*
[3]*Professor, Saveetha Engineering College, Erode, Tamil nadu, India*

Email: [1]*dineshps@bitsathy.ac.in,* [2]*mageshkumar@bitsathy.ac.in,*
[3]*dr.g.nalinipriya@gmail.com*

Abstract. The healthcare sector is one of the most challenging and fastest growing sectors in research issues. One of these is medical care. The hospital management needs to control the process of report generation and distribution of reports to the user (patient or doctor). The process is done manually. We propose a project that develops an automated medical report-ing service for health care environment. This is achieved by creating two window services. Initially the window service1 checks the status field of all rows in the transaction table and selects the rows for which the reports should be generated where the status field is "OPEN". For each row selected, it browses through the master tables and gets the appropriate stored procedure and parameters to execute them and get the data which are required to generate the report. These data are stored in the data sets and they are connected to the RDLC file where the design of report is specified. A folder is been created named with the user-id in a specified location and the generated medical report is saved into the created folder as a PDF file. The window service then updates the states of the field as "WIP" (Work in process). The Window service2 also checks the status field and selects the rows for which email should be sent, where the status field is "WIP". It fetches the appropriate PDF file which is saved by window service1 and sends it as e-mail to the user, where the E-Mail ID of the user is re-trieved from database. Finally, when the email is sent it updates the status as "CLS" that represents close. Each and every action performed by both the Window Services is recorded in the Event Log. It is very useful see the happenings in the window services and also, we can save it as a file if needed.

1. INTRODUCTION

The healthcare sector is one of the most challenging and fastest growing sectors in India [1]. Revenues from the healthcare sector account for 5.2 per cent of the GDP, making it the third largest growth segment in India. This project comes under one of the health care sectors called medical care sector.

1.1   Problem statement

There are various hospitals to which people go in for treatments. The medical report should be generated and given to the patient [2]. The patient needs to wait for a long time to get the reports when there is a greater number of patients. This is not convenient always. Nowadays in the challenging world, waiting time is one of the important in order to reduce the time spent by the patients in the hospital for their reports, we propose a system known as "Explo-sion system in medical field" [3]. Here we generate the reports automatically that are neces-

sary for the users (patient or doctor) according to the details they specified in the database [4]. Our research objective is to propose a system that provides an automatic medical report generation in the medical field. It is evident that such process could only be done by considering first the window service. In fact, as we mentioned above, we are going to create two window services for report generation. The necessary details are filled in the database as a reference for the window services [5]. The window service1 is used to fetch the required medical data from the database and generate the necessary reports. The data fields which are required for the generation of medical reports are got by the result of executed stored procedures 2 Those are specified in the database. The fetched data are stored in the form of datasets. RDLC files are created according to the type of report and connected to the appropriate datasets. Thus, while executing, final reports are generated. The generated report is stored in specified location as a PDF file. The Window service2 is used to retrieve the generated report from the location where the window service1 has saved [6]. The report that is retrieved from the destination path as a PDF file is attached and sent to the requested user as Email. The Email ID of the particular user is also fetched from the database.

2. RELATED WORKS

2.1. Window service
A Window service is a neat way to automate task that need to be performed on a predefined interval[7]. Using Visual Studio, we can create a window service and automate it. A window service is a long-running executable that performs specific functions and which is designed not to require user intervention. Windows services can be configured to start when the operating system is booted and run in the background as long as Windows is running, or they can be started manually when required problems faced by the people.

Once a service is installed, it can be started or stopped in the Services tab in Windows Task Manager [8]. The windows services can be automatically started when the computer boots, can be paused and restarted, and do not show any user interface [25]. These features make services ideal for use on a server or whenever you need long-running functionality that does not interfere with other users who are working on the same computer. You can also run services in the security context of a specific user account that is different from the logged-on user or the default computer account. The Service Controller component is used to connect to an installed service and manipulate its state; using a Service Controller component, you can start and stop a service, pause and continue its functioning, and send custom commands to a service [9].

Client-side reporting architecture
The architecture of client-side reporting revolves around your client application, as shown in Figure 1-1. Client application gathers data from your favorite data source and processes the report definition to produce a report [10]. The Report Viewer presents the result to users. The interesting idea here is that all steps needed to produce a report are an integral part of the client. The steps needed to create the report definition and to collect data are identical for all supported clients. Report Designer is a collection of graphical query and design tools that are hosted within the Microsoft Visual Studio environment [11]. Report Designer provides a Report Data pane to organize data used in your report, and tabbed views for Design and Preview so that you can design a report interactively. Report Designer also provides query designers to help specify data to retrieve from data sources and the Expression dialog to specify report data to use in the report layout [12]. Report Viewer is a freely redistributable control that en-

ables embedding reports in applications developed using the .NET Framework. Reports are designed with drag-and-drop simplicity using Report Designer included in Visual Studio 2010. The control can process and render reports independently using a built-in engine (**'local mode'**) or it can display reports that are processed and rendered on a Report Server (**'remote mode'**).

3. PROPOSED METHODOLOGY

Our research objective is to propose a system that provides an automated medical report generation and email service [24]. This is achieved through two window services [13]. Where in, all the reports required are generated and sent as a PDF file to the particular patient"s email-id. Initially the data which are required to generate the report and the details of the patient are stored in the Once the Window services are started they read the status of every record presents in the transaction table [14]. If the status is "OPEN" the first window service generate report and saves it as a PDF file in specified location. After the report is generated and saved it updates the status as "WIP" that is Work in process. If the status is "WIP" then the second window service fetches the appropriate PDF file and sends it as an e-mail to the user. When the email is sent it updates the status as "CLS" that is close.

Module Description
A Window service is a neat way to automate task that need to be performed on a predefined interval [15]. Using Visual Studio, we can create a window service and automate it. In this project we are using tow window services**.**

Window Service1:
The Window service 1 is used to generate the medical reports that are required to the users and store it as the PDF file in a specified location. The service refreshes for a specified interval of time [16]. Initially this window service checks the status field of all rows in the transaction table and selects the rows for which the reports should be generated for which the status field is "OPEN" [17]. Then according to the code, it browses through the master tables and gets the appropriate stored procedure and the parameters which are required to generate the particular report. This is done for each and every row which is been selected [18]. Once this process is done it executes the stored procedures where it gets the data which are required to generate the report. These data are stored in the data sets 28 and the data sets are connected to the specified RDLC filev[19]. The report design is specified in RDLC. A folder is been created named with the user id in the specified location. Then the generated medical report is saved into the created folder as a PDF file. Since PDF file cannot be edited the data will be secured. After the work is done the window service updates the states of the field as "WIP" (Work in process) representing that the medical report is not yet sent as an email.

Window service2:
The window service2 is created to send the medical reports as an attachment to the requested user via email.
When the service is started it also checks the status field of all rows in the transaction table and selects the rows for which the email should be sent, where the status field is "WIP" [20]. Then the window service fetches the appropriate PDF file which is saved by window service1 database in the form of tables [21] [23][26]. We create Window Services which refreshes in a specified interval of time. the user to which the report should be sent will also be retrieved from the master tables in the database. Finally, when the email is sent it updates the status as

"CLS" that is close. Each and every action performed by both the Window Services is recorded in the Event Log [22]. It is very useful see the happenings in the window services and also, we can save it as a file if needed.
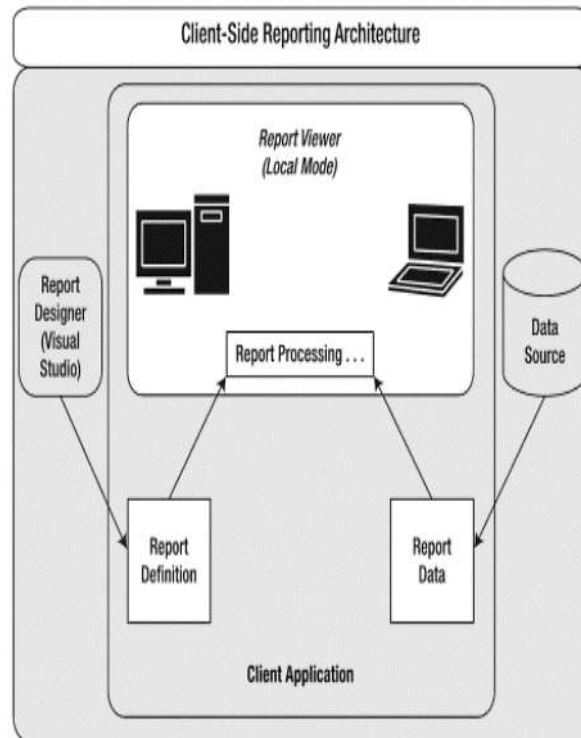

Fig 1. Client side reporting architecture

4. RESULTS AND DISCUSSIONS

To evaluate the proposed system, a suitable and standard environment is needed. We have successfully implemented the proposed system under Visual studio 2010 and SQL server 2008. The database is taken care by the SQL server and the windows service designed by visual studio. There are about 1000 reports generated for different disease case studies. The figure 9.1 and 9.2 explains performance of the proposed KMES system and existing system. We have compared the proposed KMES system with the existing system in the aspects of three major performance factors. Those factors include Time consumption, Manual intervention and Quick reply. This comparison is showed in the form of graph in the figure 2. Creation using visual studio IDE 2010. The details required for the window services were filled in the and sends it as an e-mail to the user.
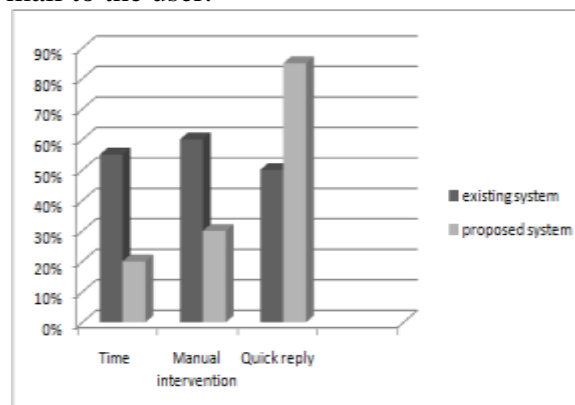
Fig 2. Performance factors of Existing system and proposed system

The figure 2 explains that the proposed KMES system is less time consuming less manual intervention and Quick reply when compared with the existing system. It is clearly shown that the proposed system gives 90% quick reply compared with the existing system. The proposed KMES system has more advantages compared with the existing system. This comparison is done based on three factors such as reliability, latest trend appliances and accuracy.
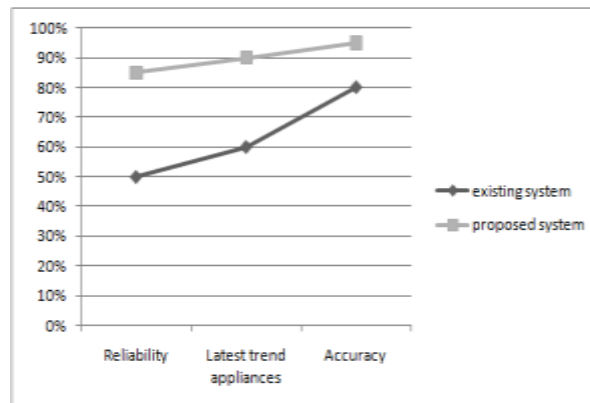

Fig 3. Merits of Existing system and proposed system

The figure 3 clearly shows that the proposed KMES system is more reliable, accurate and applies a latest trend that evolves in the market. Almost 90% latest trend is applied in the proposed system that includes Window services, automatic report generation and sending report as an email.

5. CONCLUSION

We have presented an automated report generation and services for health care environment. master and transaction tables using Sql server 2008. WS1 was created to fetch the details from the database and generate the reports. Thus, the complete sets of data were executed using stored procedure. Such those data were filled in the dataset for the report designer to generate the report. The final report was stored in the PDF format at the certain location for future access. WS2 was created to retrieve the generated report and sent to the user through Email. Each and every action performed by the Window Services was recorded in the Event Log. It was very useful to see the happenings in the window services.

6. REFERENCES

[1]. Victor r. Prybutok , amanda spink,(1999) " Transformation of a health care information system: a self-assessment survey" ieee transactions on engineering management, vol. 46, no 3.
[2]. paolo romano and francesco quaglia (2011) "Providing e-transaction guarantees in asynchronous systems with no assumptions on the accuracy of failure detection" Ieee transactions on dependable and secure computing, vol. 8, no. 1.

[3]. Ashish kamra and elisa bertino, fellow (2011) "Design and implementation of an intrusion response system for relational databases" ieee transactions on knowledge and data engineering, vol. 23, no. 6.

[4]. Jinsong liu, dong liu, xiaolu li, hesen liu, , and peng mao (2011) "Study on data management of fundamental model in control center for smart grid operation" ieee transactions on smart grid, vol. 2, no. 4.

[5]. Niilo saranummi, fellow (2008) "In the spotlight: health information systems. ieee, ieee reviews in biomedical engineering" vol. 1.

[6]. christos a. Frantzidis, charalampos bratsas, manousos a. Klados, evdokimos konstantinidis, chrysa d. Lithari, ana b. Vivas, christos Papadelis, eleni kaldoudi, eee, costas pappas, and panagiotis d. Bamidis,(2010) "On the classification of emotional biosignals evoked while viewing affective pictures: an integrated data-mining-based approach for healthcare applications" Ieee transactions on information technology in biomedicine, vol. 14, no. 2.

[7]. sang-jin lee, omid kavehei, yoon-ki hong, tae won cho, younggap you, kyoungrok cho and kamran eshraghian(2010)"3-d system-on-system (sos) biomedical-imaging architecture for health-care applications" Ieee transactions on biomedical circuits and systems, vol. 4, no. 6.

[8]. This was achieved through two windows services

[9]. Fred r. Beyette, jr., gerald j. Kost, charlotte a. Gaydos, bernhard h. Weigl (2011) "Point-of-care technologies for health care" ieee transactions on biomedical engineering, vol. 58, no. 3.

[10]. Ann séror, eresearch collaboratory,quebec city, pq, canada. (2011) "Design of virtual infrastructures for public and private services: the indian health care system" Proceedings of the 44th hawaii international conference on system sciences.

[11]. Atam p. Dhawan, senior editor in-charge, bruce c. Wheeler, editor-in-chief, ieee tbme. "Introducing tbme letters special issue on emerging technologies in point-of-care health care" Ieee transactions on biomedical engineering, vol. 58, no. 3.

[12]. "World health organization, medical devices: managing the mismatch" an outcome of the priority medical devices project. Geneva: world health organization, 2010.

[13]. Marion souil and abdelmadjid bouabdallah "On qos provisioning in context-awarewireless sensor networks for healthcare"

[14]. Wei wang, honggang wang,michael hempel, dongming peng,hamid sharif and hsiao-hwa chen (2011) "Secure stochastic ecg signals based on gaussian mixture model for e-healthcare systems" Ieee systems journal, vol. 5, no. 4.

[15]. M. J. Suárez-cabal, c. De la riva and j. Tuya(2010)"Populating test databases for testing, sql queries" Ieee latin america transactions, vol. 8, no. 2.

[16]. j. Tuya, m.j. Suárez-cabal, c. De la riv(2009) "full predicate coverage for testing sql database queries", software testing, verification and reliability,

[17]. M.j. Suárez-cabal, j. Tuya(2009) "structural coverage criteria for testing sql queries", journal of universal computer science, vol. 15, no 3, pp. 584-619

[18]. gwénolé quellec, mathieu lamard, guy cazuguel,christian roux, fellow and béatrice cochener(2011) "Case retrieval in medical databases by fusing heterogeneous information" ieee transactions on medical imaging, vol. 30, no. 1.

[19]. luis tari, phan huy tu, jo¨ rg hakenberg, yi chen, tran cao son, graciela gonzalez, and chitta baral(2012) "Incremental information extraction using relational databases" ieee transactions on knowledge and data engineering, vol. 24, no. 1

[20]. T. Dasu and T. Johnson, Exploratory Data Mining and Data Cleaning. Wiley Series in Probability and Statistics, 2003.

[21]. mohammad farhan husain, james mcglothlin, mohammad mehedy masud, latifur r. Khan, and bhavani thuraisingham, fellow (2011) "Heuristics-based query processing for large rdf graphs using cloud computing", ieee transactions on knowledge and data engineering, vol. 23, no. 9.

[22]. I. Bratko, Prolog (2001) "programming for artificial intelligence", 3rd ed. Harlow, England; New York: Addison Wesley.

[23]. proc. Acm sigmod int"l conf. Management of data (sigmod), y. Xu, p. Kostamaa, and l. Gao, (2010) "integrating hadoop and parallel dbms,"

[24]. Sujatha, K & Shalini Punithavathani, D 2016, 'Local and Global Tone Mapping Operators in HDR image processing with amalgam technique, IJEAT, ISSN: 0976-3945, Vol. 8, No.1, pp. 476-485.

[25]. Basha, A.J., Balaji, B.S., Poornima, S. et al. Support vector machine and simple recurrent network based automatic sleep stage classification of fuzzy kernel. J Ambient Intell Human Comput (2020)

[26]. S. Ramamoorthy, G. Ravikumar, B. Saravana Balaji, S. Balakrishnan, and K. Venkatachalam, "MCAMO: multi constraint aware multi-objective resource scheduling optimization technique for cloud infrastructure services," *Journal of Ambient Intelligence and Humanized Computing,* pp. 1-8, 2020.