Towards Developing Test Case Plan for Medical Report Management System (MRMS)

Norita.†, Firdaus.†, Johanna.‡

†Faculty of Computing, Universiti Malaysia Pahang, Pahang, Malaysia
‡School of Computing, Universiti Teknologi Malaysia, Skudai, Malaysia

*Corresponding author: eta305@yahoo.com.my

Abstract. Software testing is known as one of the main activities in the software development life cycle that can cause system failure without a proper plan. The testing phase consists of a few activities, such as developing the test case plan, the test design, and the execution and evaluation process. Medical Report Management System (MRMS) is one of the subsystems of the Hospital Management System that has been proposed to overcome all issues when using the manual process. The MRMS helps in controlling and monitoring the hospital records and this can affect the hospital performance. Due to that reason, a proper and well plan test case plan is needed to make sure that MRMS execute as what has been proposed by the hospital staff. This project proposed a test case plan for the testing process of the MRMS. The test plan will be used for the testing processes and the results of the testing phase will be reported in future work. The test plan will be used to plan the testing phase wisely and provide solutions in order to alleviate common random testing issues. This MRMS has different characteristics that need to be verified. If it's not tested as a whole, the users may find that the system becomes very slow or may malfunction when applications for medical reports are high at one time.

1. Introduction

Medical Report Management System (MRMS) is an efficient records management that enables effective management in delivering services. Every hospital receives many patients and this increases day by day. Some patients require their medical records for a variety of purposes, such as health insurance claims, employment requirements, courts requirements, and hospital medicine prescription with different health clinics. For these purposes, patients, close relatives, agents, or attorneys may request medical reports from the Medical Records Department through the system. They will be provided with a reference number and any query regarding their health report applications can be made using the system.

Most hospitals still use manual forms for the applications of medical reports. This method continues to pose a lot of setbacks and problems to medical practitioners, nurses, patients, and other staff in hospitals (C., C., & E., 2018). When the customer submits the form with a complete application checklist, the clerk will assign the report searcher to retrieve the patient’s medical report file. Subsequently, the assigned doctors or specialists in charge will prepare the medical report. This preparation duration of such a medical report can take up to 30 days from the date that the complete application has been received. Lastly, the Assistant Medical Record Officer will generate the report needed for the hospital management.
The MRMS allows for more organised applications of health reports, which will be accessible to both parties, i.e. hospitals and patients, compared to the old system using manual forms to be filled out and sent to the counter. Hence, this system which is to be managed by the Medical Report Department will be responsible for processing the medical reports as requested, as well as analysing, and providing statistics on hospital care. Consequently, this report will explain the related technologies of the developed system, functions, and requirements of the system, implementation, and the testing plan. Based on the system’s characteristics, the proposed framework of the testing process software is designed so that it suits testing on a client-server application or web-based system, which can be implemented in the future.

2. Related Work

Software Testing is the process of checking the correctness, completeness, and accuracy of a developed computer software (Uddin & Anand, 2019). In other words, it is a task to check whether the actual results match the expected results. By using automation tools or manual procedures, software testing helps to identify errors, gaps, or missing requirements compared to the actual requirements of the software’s components. (Sulaiman & Kassim, 2011) proposed a Customized Software Testing Process (CSTP) for banking applications, such as the Shared Banking Services (SBS) system, with regard to the software development phases. The CSTP consists of 5 phases: Business Requirement, Formal Specification, Architecture Design, Detailed Design, and Code and Implementation. During the requirement phase of the project, Test Plan documents should be prepared earlier, to identify the best test strategy and approach for further software testing process. Test Design during the specification phase is important to be conducted based on whether manual or automated testing is done. Test cases specification during the detailed design phase should be organised based on positive or negative scenarios, boundary conditions, or real-world scenarios that are usually collected into several test suites. Next, the test execution by the tester can begin as early as possible when the modules designed in the test cases are available. Then, the test results need to be evaluated until escalating problems that arise can be resolved; the test results are also documented.

Many modules involved in the system need to be tested in the Online Shopping System (Ye, Liu, & Jiang, 2014). In the user registration test, users’ name, gender, password, and address are to be input. In the order management test, the administrator is to add, query, modify, and delete order information, and ensure the accuracy and security of input information to prevent errors from information overload. In the shopping cart management test, the shopping cart function is tested for the correct implementation. Thus, in the Online Shopping System, the response time optimisation, page loading speed optimisation, and connection time are tested for efficiency.

According to the Hospital Database Management System (HDMS) for Medical Doctors, the development of this system was carried out following the procedures for developing a system using System Development Life Cycle (SDLC) (C. et al., 2018). For testing and evaluation, the framework of the entire system was set up with a module in the initialisation phase. After testing to verify accuracy, subroutines were added. Thus, the systematic top-down testing was done before the rest of the larger complex routines would be added. During implementation, programmers would work on the modules in parallel and periodic testing. The whole system is able to check the performance of intricate management growth without introducing untraceable bugs. This involves user training for operating the new information system.

3. Related Technology

3.1. Hypertext Pre-Processor (PHP)

PHP is used in this system as a programming language, which is a popular programming language for web application development (W3Techs, 2020). In addition, PHP has also recently been used in web frameworks, such as CodeIgniter, Yii, and Symfony2 (Prokofyeva & Boltunova, 2016). The PHP language is highly dynamic and provides flexibility to programmers, allowing for a fast and stable development pace (Hills, Klint, & Vinju, 2013; Prokofyeva & Boltunova, 2016). Thus, PHP is an appropriate language for the MRMS. The PHP programming uses the Model-View-Control (MVC) framework. The MVC has been proven to simplify the development process to be more efficient...
(Hardono, Isti Surjandi, Amar Rachman, Yantie Arsita Br. Panjaitan, 2017). Fig. 1 describes the MVC framework for the MRMS system.

![Figure 1. The MVC framework](image)

3.2. **Structured Query Language (MySQL)**
For data storage, MySQL is used to store and access data quickly and securely (Ansari, 2018). MySQL is also lightweight compared to other databases (Ye et al., 2014), and contains several tables to store information that is mainly used in the system, such as Patient Profile, Applicant Profile, Application Details, and others.

4. **System Functions and Requirements**
4.1. **System Functions**
MRMS is a system developed to handle applications received by the Medical Records Department (MRD) in hospitals. MRD is the department responsible for managing the applications of medical reports. These applications are for personal, lawyer, police, insurance, or government’s perusal to be used for insurance claims or other legal purposes. Generally, every hospital would have this department, which receives many applications. Managing this application manually is a cumbersome task and numerous mistakes can potentially occur, which can be detrimental to many parties.

Therefore, this system is developed for medical records officers to monitor such requests routinely and to ensure that each application is processed efficiently. All applicants may also apply medical records electronically and check the request status online. It is also possible to monitor and track any payment for medical records through this system. Doctors will also be alerted to new applications by e-mail notification, saving time and reducing the time needed to prepare the medical records. Fig. 2 shows the system function graph that has been proposed for the MRMS.

![Figure 2. System Function Graph MRMS](image)
4.2. System Requirements

MRMS is an online system developed to facilitate the application of records via a web-based platform. Taking into account the problems of manually managing applications for records, most can be resolved with the suggestion of using an online system. This can decrease large crowds of applicants in the MRD. The online system runs on Windows Server, using XAMPP web server. The design of the system will be established after analysis of the functional requirements has been carried out. First, the system provides an online application form. This form gathers all the requirements needed to process the application. The entire process will be updated, including the payment and application status of the requested medical report. The next section involves the MRD user interface, which the user needs to log in. The interface and menu are provided based on their roles in the system. Finally, the system will provide a report generation module needed for statistical purposes.

5. Implementation

Fig. 3 below shows the general view of the flow for the MRMS. Customers or patients will apply the medical report from the system after all required payments are settled. The MRD will process the request, assign the report searcher, and doctors in charge before sending back the completed medical report to the customers.

![Figure 3. General view of MRMS](image)

The widely-used Model View Controller (MVC) architecture is implemented in the web application development. The MVC enables developers to work in parallel with different components without affecting or blocking one another (Selfa, Carrillo, & Del Rocío Boone, 2006). As illustrated in Fig. 4, the MVC divides the interactive system into three components, with each one specialising in a specific task. The model contains the application data and manages the core functionality. The view manages the visual display of the model and feedback to the user. The controller interprets the input from the user, commanding the model and the view to change appropriately.

![Figure 4. MVC Architecture](image)

This system runs on Windows operating system using MySQL database that has a good performance in operation management. The MySQL database system is mainly used for storing all kinds of related information needed for the system (Ye et al., 2014). The main function modules of the system include preparing medical reports, updating information of the reports, and generating the completed reports. The module for preparing the medical report allows the clerk to assign the name of the report searcher by referring to the report number to be retrieved from the brackets. Afterward, receipt number and specific
date must be keyed in before being sent. Details of the payment receipt and additional notes can also be added if necessary. In the module that updates the report’s information, the clerk can assign doctors or specialists depending on the medical report requested. The clerk also will be able to update the report’s completion date, as well as the status of the report, whether it has already been submitted or is still being processed. In the module that generates the report, the user produces the completed report and may convert it into Microsoft Excel format.

6. Testing Plan
Unit testing has been chosen to evaluate the system’s compliance with the requirements given by the stakeholders and customers. Faults are intended to be revealed and units are to be identified and tested for possible points of failure. The testing process consists of several steps: identify the order of unit that should be tested; create a list of test cases with possible value combinations; execute the testing phase; and gather results and analysis (Zhang, Thangarajah, & Padgham, 2007). In this project, the test case plan is documented in the Software Testing Design (STD), while results are documented in the Software Testing Result (STR).

6.1. Test Design Architecture
This project applies the Model View Controller (MVC) architecture. The MVC has been used widely in application development that allows it to work in parallel with different components. Fig. 5 shows the test design architecture for the MRMS.

![Figure 5. Test Design Architecture](image)

6.2. Hardware and Software Preparation
Testing of the project will be performed on the development platforms. No special hardware preparation is required. The designated machines must be operational and running.

i. The hardware to be used for testing is a Windows Server 2003, equipped with an operating platform consisting of PHP programming language.

ii. The system must include an Ethernet cable and power supply cables.

The software component requirements for this project are as follows:

i. Virtual Sever – PROXMOX VS 6.0
ii. Operating System – Windows Server 2003
iii. Web Server – XAMPP 7.3
   a. Programming Language – PHP
iv. Database – MySQL
6.3. Test Description

Most hospitals face issues and challenges with their Medical Records Department. Such challenges are prevalent since some staff still prefer to execute business activities using manual processes (Olamide, Adedayo, & Abiodun, 2015), as they mistrust computerised systems due to long-standing familiarity with manual processes. Due to this reason, the test case plan must be designed very carefully in terms of the conditions and variables to meet the system requirements’ specification. Table I shows an example of the test case description for “Status Incomplete Document.” “Status Incomplete Document” is one of the events in the MRMS. The test description explains the purpose of the test while the precondition expresses a function’s expectation on its arguments. Two test inputs are expected for this test case: the first input anticipates that the user will click the ‘save’ button without checking any checkbox, while the second input is in case the user clicks the ‘save’ button without keying in the amount field for the payment section. Finally, a comprehensive result is to be expected. Upon execution, if the test case fails to produce the expected result, it is considered as failing to meet the test case objective.

| Test Case ID    | TC-DI-02                  |
|-----------------|---------------------------|
| Description of test | To verify Save button |
| Precondition    | 1. Login successfully as User |
| Test Input      | 1. Click button with empty checkbox |
|                 | 2. Click button with tick on payment needed checkbox without filling in the amount |
| Expected Test Results | 1. A “Please tick incomplete document required” error message will be displayed |
|                 | 2. A “Please fill in Payment Amount” error message will be displayed |

This is an ongoing research, which is currently in the process of planning the test case plan by referring to the scenarios of one public hospital. Once the test case execution is done, results will be analysed and documented in the Software Testing Result (STR). The STR will include the status of test and quality targets upon completion of the implementation stage. The STR can assist stakeholders to make critical decisions, especially those related to the tested version. Furthermore, it will summarise the test status, such as the number of test cases planned to be completed, the number of test cases to be executed, and the status report of test incidents. One example of an incident report is that the system does not display any error message when the applicant inserts incorrect information during the application process. The test coverage and result consist of information on the test cycles, which will help to produce a significant value for the testing activities.

7. Conclusions

Software testing is one of the phases in the life cycle of software development that is widely used to validate and verify whether or not a product meets the system requirements. The testing phase is a well acknowledged costly phase. Thus, the test case needs to be planned wisely so that testing can ensure the system will meet the requirements and can handle any possible input to be entered by the user. This project attempts to design a proper test case plan to assess the Medical Records Department System. The results of the test, based on the proposed test case plan, will be used in the next stage of the project for further improvement. System quality and reliability of the proposed system will be into
considerations. Besides, the proposed test case plan can also be used to measure the efficiency of the system especially in terms of system functionality and responsiveness. A preventive approach, such as the proposed test plan and the test design, is highly crucial in producing test cases that can detect faults as early as possible.

References
[1] Ansari, H. (2018). Performance Comparison of Two Database Management Systems MySQL vs MongoDB. [2] C., A. J., C., A. V., & E., N. S. (2018). Design and Implementation of a Hospital Database Management System (HDMS) for Medical Doctors. International Journal of Computer Theory and Engineering, 10(1), 1–6. https://doi.org/10.7763/ijcte.2018.v10.1190
[3] Hardono, Isti Surjandi, Amar Rachman, Yantie Arsita Br. Panjaitan, A. R. (2017). Development of Theses Categorization System Search Engine Using PHP and MySQL. International Conference on Information Technology Systems and Innovation, 3–8.
[4] Hills, M., Klint, P., & Vinju, J. (2013). An empirical study of PHP feature usage: A static analysis perspective. 2013 International Symposium on Software Testing and Analysis, ISSTA 2013 - Proceedings, 325–335. https://doi.org/10.1145/2483760.2483786
[5] Olamide, O., Adedayo, E. W., & Abiodun, O. a. (2015). Design and Implementation of Hospital Management System Using Java. IOSR Journal of Mobile Computing & Application, 2(1), 2394–42. https://doi.org/10.9790/0050-0213236
[6] Prokofyeva, N., & Boltunova, V. (2016). Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems. Procedia Computer Science, 104(December 2016), 51–56. https://doi.org/10.1016/j.procs.2017.01.059
[7] Selfa, D. M., Carrillo, M., & Del Rocio Boone, M. (2006). A database and web application based on MVC architecture. Proceedings of the 16th IEEE International Conference on Electronics, Communications and Computers, CONIELECOMP 2006, 2006(March 2006), 48. https://doi.org/10.1109/CONIELECOMP.2006.6
[8] Sulaiman, N. A., & Kassim, M. (2011). Developing a customized software engineering testing for Shared Banking Services (SBS) System. Proceedings - 2011 IEEE International Conference on System Engineering and Technology, ICSET 2011, 112–137. https://doi.org/10.1109/ICSEngT.2011.5993436
[9] Uddin, A., & Anand, A. (2019). Importance of Software Testing in the Process of Software Development. LISRDI-International Journal for Scientific Research & Development, 6(January), 2321–0613.
[10] W3Techs, (2020, January 7). Usage Statistics and Market Share of Server-side Programming Languages for Websites. Retrieved from https://w3techs.com/technologies/overview/programming_language
[11] Ye, G. Z., Liu, X. W., & Jiang, W. X. (2014). Design and implementation of online shopping system based on MVC frameworks. Applied Mechanics and Materials, 687–691, 2466–2469. https://doi.org/10.4028/www.scientific.net/AMM.687-691.2466
[12] Zhang, Z., Thangarajah, J., & Padgham, L. (2007). Automated unit testing for agent systems. ENASE 2007 - Proceedings of the 2nd International Conference on Evaluation of Novel Approaches to Software Engineering, 10–18. https://doi.org/10.5220/0002585900100018