

Invigilator Assignment based on District Module

Chuan Wei Jian and Noraini Ibrahim

School of Computing, Faculty of Engineering
University Technology Malaysia
81310 Johor Bahru, Malaysia

wei.jian@graduate.utm.my, noraini_ib@utm.my

Abstract—Exam invigilators is responsible for supervising the candidates of exams throughout the whole time of examinations to ensure the fairness of the exam result. One of the pre-works that need to be done before the examination can be carried out is the exam invigilator assignment. However, the process of assigning exam invigilators is time-consuming and tedious. The problem worsens when the officer responsible for assigning the exam invigilator in one district is limited. The project goal of developing the Invigilator Assignment based on District Module for Malaysia Public Exams Invigilator Assignment System is to improve the process of assigning invigilators to exam centers within a district by introducing the automation of assigning invigilators and compiling formal letters. The developed Invigilator Assignment based on District Module provides the five main features which are: (i) automatic assigning of the invigilator, (ii) management of assignment task, (iii) automatic compiling and sending of formal letter, (iv) management of letter template and (v) management of exam center's information. Agile development methodology is selected with some recent technologies such as React.js, Node.js and MongoDB implemented in this project. During the testing phase of Invigilator Assignment based on District Module, Black box testing and User acceptance testing are selected to validate the module's functionalities.

Public Examination, Invigilator Assignment System, web application, Agile, MERN

I. INTRODUCTION

Exam invigilators assignment is one of the necessary pre-work that needs to be done before the examination can be carried out, especially when it is related to public examination in Malaysia such as UPSR, SPM, and STPM. Invigilators are responsible for supervising exams through some form of surveillance, ensuring that with their presence students are reluctant to engage in cheating on one another or through other prohibited means [1]. The assignment process of invigilators to exams could be a very difficult and time-consuming process when there are too many exam centers and invigilators who will be assigned [2]. The assignment of exam invigilator is complicated even more by the different roles of the exam

invigilator required for each exam center. Apart from the complexity of the work of invigilators assignment, the usage of paper documents during the assignment process is tremendous. In Johor, there are 46 types of public examinations needed to be handled by all the Pejabat Pendidikan Daerah (PPD) from different districts in Johor. The Jabatan Pendidikan Negeri Johor (JPNJ) then decided to use an invigilator assignment system built from Microsoft Access named ExamPPD to solve the issue of paper documents and the manual assignment process.

However, there are still some other issues that remain even after the usage of the ExamPPD which are the extra process of extracting collected data from the exam centers and import into the ExamPPD, the shortage of human resources in PPD to handle the assignment and preparing the formal letters for invigilators manually and also the lack of the ability to access the system remotely from anywhere.

The process of collecting and extracting necessary data is quite tedious as the process is done manually and redundantly whenever conducting a public examination. Even after the assignment of the invigilators, the staff of PPD needs to compile different kinds of formal letters to be sent to the invigilators according to their roles during the exam invigilation. The invigilators with the same role will receive the letters with the same contents but just slight differences in personal information and the exam center where they will perform their duties. Hence, it is quite ineffective and inefficient to do it manually by the staff as most of the information can be obtained from the data in the system. Usually, the whole process takes about 1 to 2 weeks as there will be only 2 or 3 staff responsible to handle all the work. To solve the issue of the ineffective and inefficient assignment process, this is where an automated system comes in. System automation can greatly improve office efficiency and has great application value [3].

The aim of this study was to elicit and analyze the requirements for a new web application for the efficient and effective assignment of the invigilators, to develop a web application that shares a common database with the other two modules which allows it to directly access the information collected from exam centers without any manual extract and import action needed, to develop the web application that automates the process of assigning of invigilators and compilation of formal letter based on the assignment result.

II. LITERATURE REVIEW

This section will cover the comparison between the developed system and other similar systems before developing the system. There does not exist a system or application that consists of the functionality of assigning exam invigilator, hence the comparison is done by comparing systems with similar functionality which are the role assignment and also letter management. It is very important and paramount to compare the existing similar system in the market to help to figure out the potential weakness and problems that can be used to improve the Invigilator Assignment based on District Module. Besides, the strength of the similar systems can be adopted in the developed system to further enhance the uniqueness of Invigilator Assignment based on District Module.

A. Orangescrum

Orangescrum is a project management tool. It is accessible from both web and mobile application platforms. The Orangescrum consists of the user role management feature which will be used by the project manager to precisely and accurately distribute the task and work to each role.

B. Discord

Discord is a communication platform accessible through web, mobile applications and also a desktop application. This communication platform provides the features to create a community where it allows users to invite other users into the community to communicate through instant messaging, voice messaging or video calls. Inside the community, every member may be assigned a role and each role will be given different permission to access the features in the community.

C. MoversSuite

MoversSuite is a document management software specially designed to manage documents involving moving information. The main features provided by the software are order management, dispatch and planning. Inside the MoversSuite software, there is a Letter Management system which allows users to create document templates containing the MoversSuite order data [4].

Table I below shows the comparison of existing software in the market with the ExamPPD system in terms of role assignment.

TABLE I. ROLE MANAGEMENT FEATURES COMPARISON

Features	<i>Orangescrum</i>	<i>Discord</i>	<i>ExamPPD</i>
Platform provided	Available in the web application and mobile application	Available in the web application, mobile application and desktop application	Only available in desktop application
Online database	Supported by online database	Supported by online database	No online database support and require manual export and import of data
Default user role	Default user role - Owner, Admin, User, Client	Default user role - @everyone	Default user role - Chief of Invigilator, Vice Chief of Invigilator, Environmental Supervisor, Room Keeper, Reserved Invigilator, Exam Invigilator
Self-defined user role	Enable user to self-defined a role and group the role into a role group	Enable user to self-defined a role	Not enable user to self-defined a role
Remove role	Enable user to remove a self-defined role	Enable user to remove a self-defined role	Not enable user to remove a self-defined role
Automatic assignment of role	No automated assignment of user role	No automated assignment of user role	No automated assignment of user role
The suggestion of the user for the role	No suggestion of user for certain role	No suggestion of user for certain role	Suggest user for certain role
View assignment result	Enable user to view the result of assignment and modify them	Enable user to view the result of assignment and modify them	Enable user to view the result of assignment and modify them

Table II below shows the comparison of existing software in the market with the ExamPPD system in terms of letter management.

TABLE II. LETTER MANAGEMENT FEATURES COMPARISON

Features	<i>MoversSuite</i>	<i>Discord</i>
Create document template	Feature provided	Feature provided
Copy document template	Feature provided	Feature not provided
Edit document template	Feature provided	Feature not provided
View document template	Feature provided	Feature provided
Remove document template	Feature provided	Feature provided
Generating document using template	Automatic filling the order information into the bookmarks of the template	Generate manually by following the template
Generating email using the template	Automatic filling the order information into	Feature not provided

Features	MoversSuite	Discord
	the bookmarks of the template and compile into email	
Create document template	Feature provided	Feature provided

III. METHODOLOGY

System development methodologies provide frameworks for managing development projects. They do not primarily address technical details of how software is developed but determine the organizational embedding of development and give guidelines for organizing activities. Thereby, they can be used to decrease the risk of project failure [5]. The Agile-waterfall method is selected as the most appropriate and suitable development methodology for developing the Invigilator Assignment based on District Module.

The Agile development methodology is a method where the specification, design, implementation and testing are interleaved. Agile methodologies appear to respond to the dynamic aspects of the environment that have emerged [5]. At each Sprint, all the development processes will be experienced including the testing. The waterfall development methodology is a plan-driven methodology where all of the software development processes are planned in advance. The project phases are sequentially performed and each process must be completed before proceeding to the next phase. The waterfall development methodology is highly suitable for projects where the requirements are clearly defined by the project owner or stakeholder so that the project team can easily and accurately plan the phases [6].

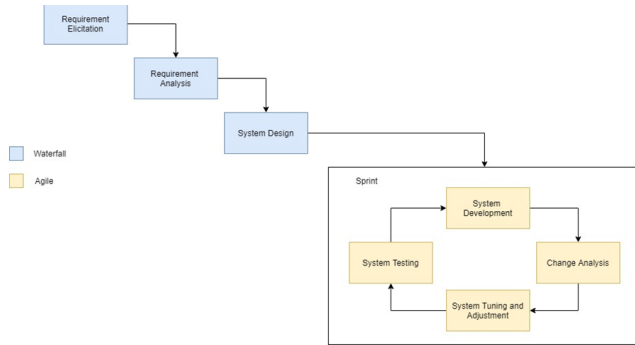


Figure 1. Agile-waterfall hybrid methodology

Fig. 1 depicts the illustration of the methodology used and each phase of the development process.

A. Requirement Elicitation & Analysis

The information and requirements of the Invigilator Assignment based on District Module were gathered from the stakeholders through questionnaire survey and interview. The respondents of the questionnaire are PPD Officer in Johor State. There are six categories of requirements which are the “attractive” (A), “one-dimensional” (O), “must be” (M), “indifferent” (I), “questionable” (Q), “reverse” (R). Attractive attributes are unexpected items for the customer and like them

when provided in the product. One-dimensional attributes that can increase customer satisfaction when met, and will dissatisfy them when missed. Must-be attributes are the features assumed by the customer to be in the product. Indifferent means the customer does not care whether the feature is provided or not, while questionable means the customer does not understand the feature. Reverse means the customer does not want the feature and they strongly expect the reverse of it [7].

Table III shows the summary of the questionnaire results. The reason that the questionnaire has fewer respondents is the number of officers that responsible for assigning invigilators in PPD is very limited, usually, one PPD only has one officer who will perform the task of assigning invigilators. The result shows that the sharing of an online database is a must and attractive requirement for the Invigilator Assignment based on District Module.

TABLE III. TABLE TYPE STYLES

Assessed Requirement	A	O	M	I	Q	R	Total	Category
Shared online database feature	1	0	1	1	0	0	3	A, M, I
Dashboard feature	0	0	0	3	0	0	3	I
Automatic invigilator assignment feature	0	0	1	2	0	0	3	I
Modify assignment result feature	0	0	0	3	0	0	3	I
View assignment result feature	0	0	0	3	0	0	3	I
Create letter template feature	0	0	0	3	0	0	3	I
Automatic compile letter feature	0	0	1	2	0	0	3	I
Modify letter template feature	0	0	0	3	0	0	3	I
Send letter through email feature	0	0	0	2	0	1	3	I

Apart from conducting a questionnaire survey, an interview was carried out with the PPD officers from PPD Johor Bahru, Encik Mohamed Nor and PPD Kluang, Encik Saifuddin through the Google Meet on 23 March 2021. The business process of assigning invigilators was clarified and the interface of SLAP and ExamPPD systems was shown. The meeting was

recorded as evidence. A use case diagram was created after the analysis of requirements from the stakeholders to depict the features and functionalities of the system. Use Case Diagram of Invigilator Assignment based on District Module

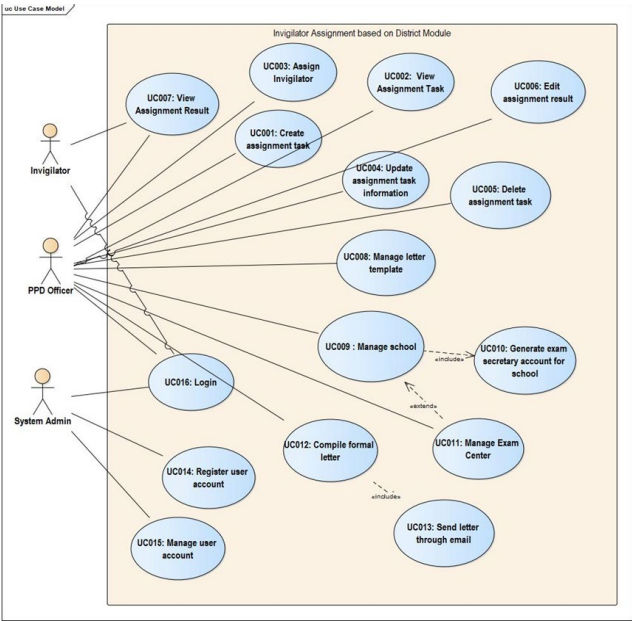


Figure 2. Use Case Diagram of Invigilator Assignment based on District Module

Fig. 2 shows the use case diagram of Invigilator Assignment based on District Module. The module consists of three main actors, System Admin, PPD Officer and Teacher.

B. System Design

System architecture design is concerned with the understanding of how the system should be organized and designing the overall structure of the system [8]. The system architecture design must be able to define a solution for realizing the functional and non-functional requirements gathered from the stakeholders. In the Invigilator Assignment based on District Module, the Model-View-Controller (MVC) is chosen due to the reason that the system built with MVC architecture is easy for maintenance, highly reusable and has less coupling of system components. There are three components in the MVC architecture design pattern which are the Model, View and Controller. Model interacts with the database system and responds to the request for data from View. View is concerned with the display of data while Controller responds to the user events and informs the Model and View when changes occurred [9].

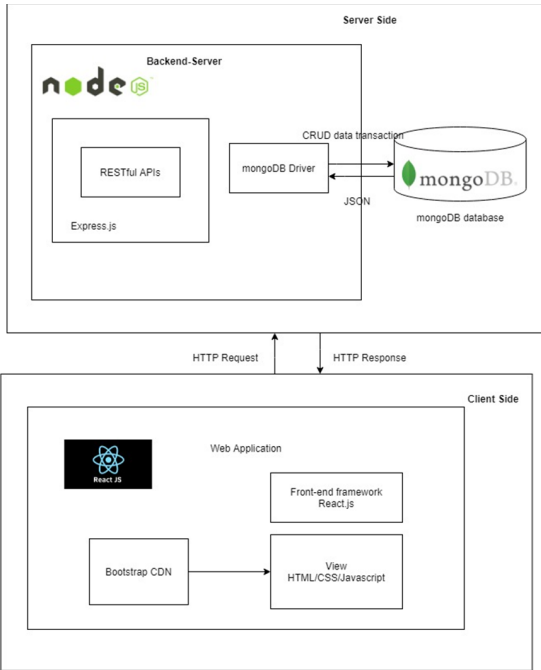
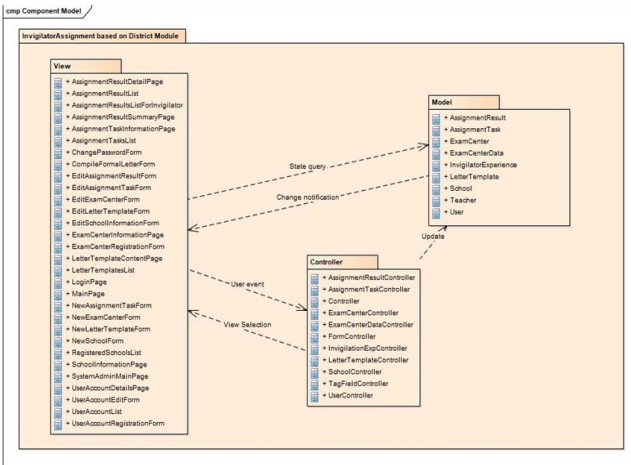


Figure 3. System Architecture Design Illustration

Fig. 3 shows the system architecture design illustration of



the Invigilator Assignment based on District Module.

Figure 4. MVC architecture package diagram

Fig. 4 shows the package diagram of the Invigilator Assignment based on District Module.

IV. IMPLEMENTATION

The module was constructed with a MERN stack which includes the NodeJS a Javascript runtime for the backend application, ExpressJS a NodeJS framework and ReactJS a Javascript frontend framework. The web application was connected to a NoSQL database which is MongoDB.

The backend of the module includes the Model and Controller components. The services of the module were implemented with RESTful API and the data were interchanged between the frontend and backend in JSON format. The request from the frontend user interface was handled by the Controller and database operations were performed by the Model. The Model consisted of the scheme that enforced the rule and ensured the data type of the data in the database.

The frontend of the application which is the View component consisted of the user interfaces. The users of the module interacted with the interface to access all the services and features provided by the module.

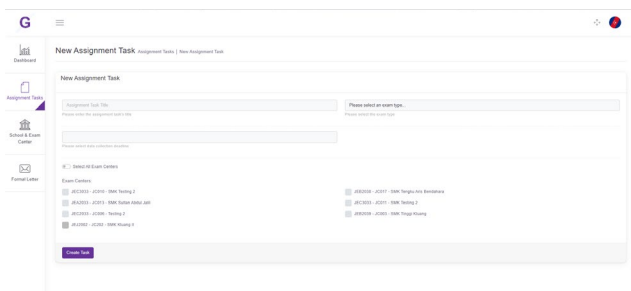


Figure 5. Create New Assignment Task Page

Fig. 5 depicts the user interface for creating a new assignment task. The PPD officer can use this page to add a new assignment task to collect all the invigilators' information needed for the public examination that will be conducted from all exam centers under the district.

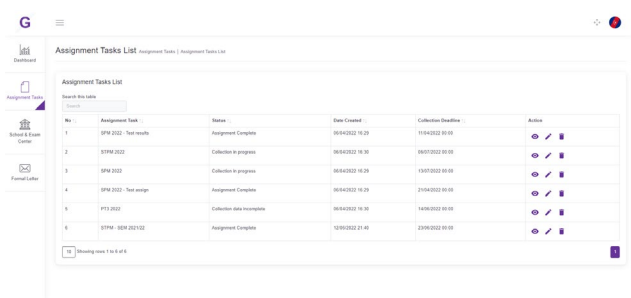


Figure 6. Assignment Task List

Fig. 6 depicts the assignment task list that enabled the PPD officer to manage the assignment task under his/her district. The officer is allowed to view, edit and delete any particular task. Each assignment task has 4 statuses which are the Collection in progress, Collection data incomplete, Assigning in progress and Assignment completed.

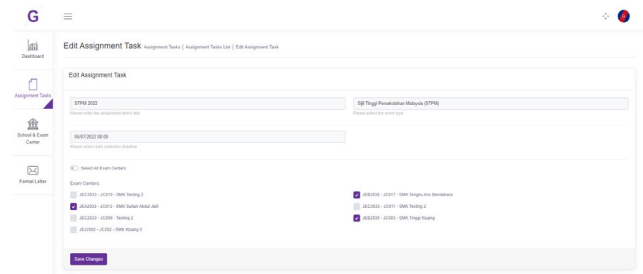


Figure 7. Edit Assignment Task Page

Fig. 7 shows the edit assignment task page which allowed the officer to change the task's information such as the deadline for collecting invigilators' information and the number of exam centers that will involve in the exam.

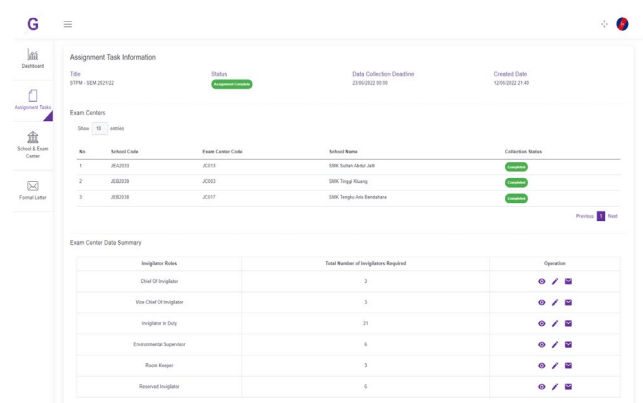


Figure 8. Assignment Task Information Page

Fig. 8 shows the detailed information of an assignment task and on this page, the officer is allowed to assign invigilators of certain roles, view the assignment result and compile letters for

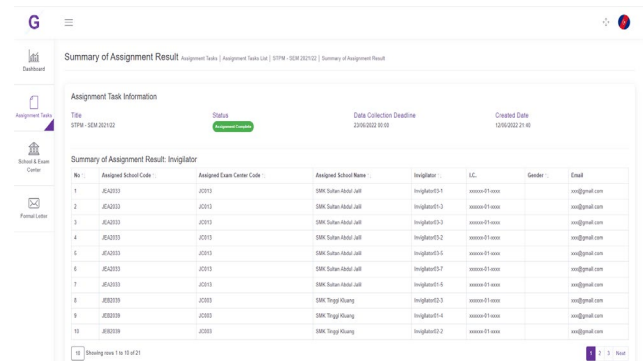


Figure 9. Assignment Result Summary Page

Fig. 9 depicts the assignment result that can be viewed by the officer once the automated assigning of invigilators has been completed.

every invigilator.

resolved to the selected field value during the compiling of the

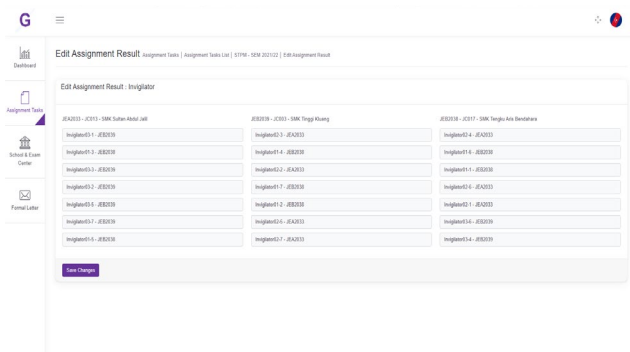


Figure 10. Edit Assignment Result Page

Fig. 10 depicts the page to edit the assignment result of any roles that the officer found that the result automated assigned by the system is not satisfied.

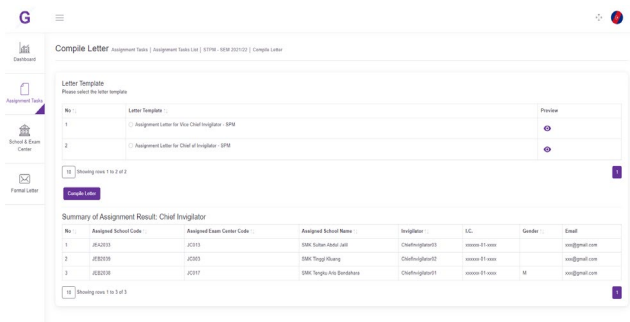


Figure 11. Compile Letter Page

Fig. 11 depicts the page that enabled the officer to select one of the letter templates created in the system and compile the letters as well as send them to each invigilator.

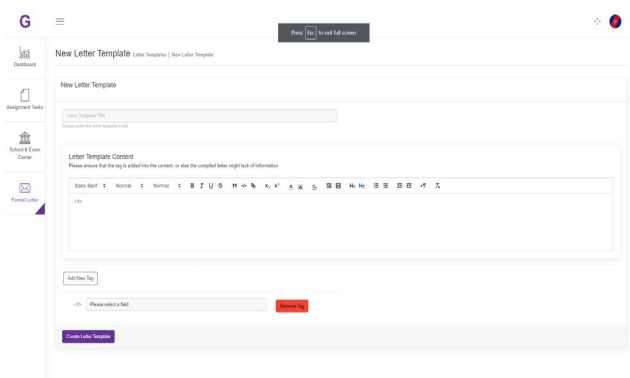


Figure 12. New Letter Template Page

Fig. 12 depicts the page to create a new letter template and the content of the letter can include certain tags that can later be



letter. For example, <1> represents the name of the invigilator, then when compiling the letter, the <1> will be replaced with the name of the invigilator.

Figure 13. Letter Template List

Fig. 13 shows the letter template list that enabled the officer to manage the letter template including viewing the content of the letter, the deleting and editing the letter template.

The module was deployed to a cloud server where the frontend application was deployed to Firebase hosting and the backend application was deployed to the Heroku cloud server.

V. TESTING AND RESULTS

A. Black Box Testing

Black box testing is the testing method that involves the input to the system and observing the output produced. The testing is aimed at the purpose of identifying the unexpected errors or bugs that may affect the robustness of the system and ensuring that the expected result is produced.

Table IV shows the summary of the black box testing that was performed.

TABLE IV. SUMMARY OF BLACK BOX TESTING

Use Cases	Expected Result	Actual Result
UC001 - Create assignment task	Success create a new assignment task	Success
UC002 - View assignment task	Success display the information of the assignment task, searching and sorting functions works fine	Success
UC003 - Assign invigilator	Success randomly assign invigilators to each involved exam center	Success
UC004 - Update assignment task information	Success edit the assignment task information	Success
UC005 - Delete assignment task	Success delete the selected assignment task	Success
UC006 - Edit assignment result	Success edit the assignment result	Success
UC007 - View assignment result	Success display the assignment result	Success
UC008 - Manage letter template	Success create, view, edit and delete letter template	Success

Use Cases	Expected Result	Actual Result
UC009 - Manage school	Success register, view and edit school under the district	Success
UC011 - Manage exam center	Success register, view and edit exam center under the school	Success
UC012 - Compile letter	Success compile letter and send them in email	Success
UC014 - Register new officer account	Success register new PPD officer account	Success
UC015 - Manage user account	Success view, edit and delete PPD officer account	Success
UC016 - Login	Success login to the system and provide authentication and authorization for the system	Success

B. User Acceptance Testing (UAT)

User acceptance testing is one of the acceptance testing methods to gather the user's feedback after interact with the developed system and also to gain enough confidence level that the developed system can perform the tasks required by the user. This section will describe the conducted UAT session on 07 June 2022.

At the beginning of the testing session, a brief explanation of the purpose and agenda of the UAT will be given to the tester. The manual will be given to the tester and he/she will follow the manual to perform the testing. The tester will be required to fill in the demographic questionnaire before proceeding with the testing. Then the session will continue with the testing and any questions, suggestion or bug report may be given at any instant. After all steps are completed, the tester will be required to fill in the System Usability Scale (SUS) survey.

Table V shows the tasks that will be performed by the tester during the UAT session.

TABLE V. TASKS OF UAT

Task	Description
1.	Manage User
2.	Manage School & Exam Center
3.	Create New Assignment Task
4.	Manage Assignment Task
5.	Assign Invigilator
6.	View Assignment Result
7.	Edit Assignment Result
8.	Manage Formal Letter Template
9.	Compile Formal Letter
10.	View Invigilator Result & History

The time taken to perform each task using the Invigilator Assignment based on District Module is recorded as shown in Fig. 14. The time taken for completing all the tasks is ranged from 1 minute to 8 minutes. This shows that the participant with the aid of Invigilator Assignment based on District Module can finish the process assigning of the invigilator in a short time. Among the tasks given, tasks such as Task 2.1, Task 2.2, Task 8.1 and Task 9.1 takes about 2 minutes due to several reasons are some tasks required filling of a lot of data field before proceeding, task to creating letter template needs some time to understand the functions of the tags while the compiling letter task needs to wait for the services to compile and send the letters to each invigilator.

Figure 14. Time Taken to Complete the UAT Tasks

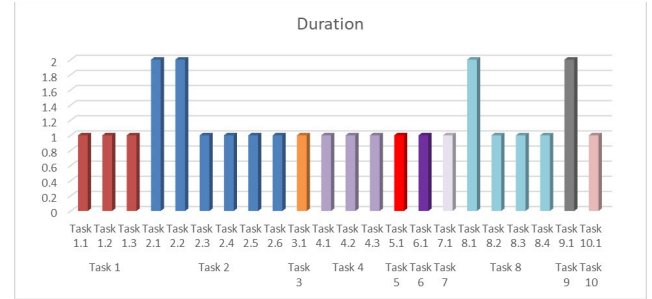
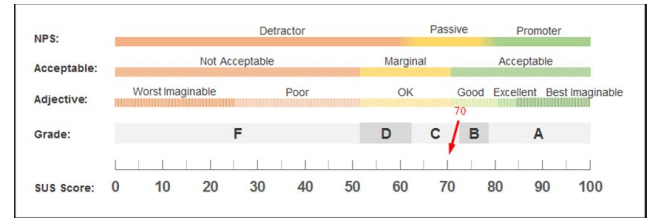


Fig. 14 depicts the time taken to complete the UAT tasks by



the tester.

Figure 15. System Usability Scale (SUS) Scale

Fig. 15 shows the SUS score of the Invigilator Assignment based on District Module. The calculation is done by following the formula:

$$(X + Y) * 2.5 \quad (1)$$

where X is the score of odd number questions added – 5 while Y is 25 – the score of even number questions. The total score of the module is 70% which is categorized as “Good” and it indicates that there are still rooms to enhance and improve the application according to the participant's suggestions.

VI. CONCLUSION

This paper provides a clear overview of the development of the Invigilator Assignment based on District Module by following the methodology selected and the outcome of each phase which includes the elicitation of requirements, analysis of

requirements and system design. This paper also gives a detailed description of the realization of the main features of the module. Finally, two tests were carried out to validate and verify the developed outcome.

From the analysis of the case study, the PPD officer always struggles with the time-consuming invigilator assignment process. The manual action of importing and exporting data from the other system to the ExamPPD system and manual preparation of formal letters is also very ineffective and inefficient in the process of assigning exam invigilators.

Hence, the overall achievement of the project is to help the PPD officer to transform the ineffective and inefficient processes into automated system processes and enable the task of assigning invigilators can be done in a short time. The processes of the automatic assigning invigilator and automatic compiling of the formal letter have been introduced to shorten the time to complete the preparation work of conducting a public examination. The online shared database is also being introduced to replace the manual import and export data action.

There are several suggestions and recommendations for the Invigilator Assignment based on District Module. The first suggestion is to have a more significant and better analysis of the data in the dashboard of the application. The current dashboard has included some presentation of data in graphs or charts. However, the data presented is lack analytics and insight. The second suggestion is to include the function to generate a report based on the completed assignment task or based on the data analytics mentioned in the first suggestion. Lastly, the application should optimize the algorithms of the randomly assigning invigilator and compiling and sending of a formal letter. These algorithms are the main function of the module, by optimizing them can directly increase the efficiency of the whole module.

ACKNOWLEDGMENT

In the preparation of this paper, we were in contact with different kinds of people from different professions. They have hugely contributed to our understanding and thoughts on the

completion of the system development, the thesis and the research paper. We express our deepest gratitude and appreciation to these people and without them, the project would not be possible to be completed in time.

We would also like to express our gratitude to our colleagues and other coursemates who provided us with the support and assistance on various occasions. We are grateful to our family members and friends that always provided us with the best encouragement and support in mind.

REFERENCES

- [1] [1] O. Mubin, M. Cappuccio, F. Alnajjar, M. I. Ahmad, and S. Shahid, "Can a robot invigilator prevent cheating?," *AI Soc.*, vol. 35, no. 4, pp. 981–989, 2020, doi: 10.1007/s00146-020-00954-8.
- [2] [2] D. KÜÇÜK MATCI and I. ACAR, "Modelling and Optimization of the Exam Invigilator Assignment Problem Based on Preferences," *Turkish J. Eng.*, vol. 3, no. 3, pp. 140–148, 2019, doi: 10.31127/tuje.467003.
- [3] [3] X. Hong, "Application and Reform of Office System Automation in Higher Vocational Colleges," no. Isaicn, pp. 129–133, 2019, doi: 10.25236/isaicn.2019.024.
- [4] [4] "MoversSuite," 2021. <http://www.ewsgroup.com/moverssuite/support/onlinehelp/2019R20/ind ex.htm#!WordDocuments/lettermanagement.htm> (accessed May 12, 2021).
- [5] [5] S. Dyck and T. A. Majchrzak, "Identifying common characteristics in fundamental, integrated, and agile software development methodologies," *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, pp. 5299–5308, 2012, doi: 10.1109/HICSS.2012.310.
- [6] [6] S. Saeed, N. Z. Jhanjhi, M. Naqvi, and M. Humayun, "Analysis of software development methodologies," *Int. J. Comput. Digit. Syst.*, vol. 8, no. 5, pp. 445–460, 2019, doi: 10.12785/ijcds/080502.
- [7] [7] B. A. Mustasfa, "Classifying software requirements using kano's model to optimize customer satisfaction," in *Frontiers in Artificial Intelligence and Applications*, 2014, vol. 265, pp. 271–279, doi: 10.3233/978-1-61499-434-3-271.
- [8] [8] Ian Sommerville, *Software engineering*, 9th ed., vol. 195, no. 4283. 2011.
- [9] [9] M. Singh, N. Chauhan, and R. Popli, "A Framework for Transitioning of Traditional Software Development Method to Distributed Agile Software Development," *IEEE Int. Conf. Issues Challenges Intell. Comput. Tech. ICICT 2019*, pp. 2019–2022, 2019, doi: 10.1109/ICICT46931.2019.8977654.