

Pre-Viva Assessment System Development for FSKTM, UTHM

Chong Zi Yong¹, Nureize Arbaiy^{1*}

¹Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

DOI: <https://doi.org/10.30880/aitcs.2022.03.02.097>

Received 24 August 2022; Accepted 02 November 2022; Available online 30 November 2022

Abstract: The FSKTM pre-viva assessment system is an assessment information system designed for postgraduate students to submit a thesis draft for assessment before proceeding with the actual viva presentation. The current pre-viva assessment system is implemented in a hybrid manual system. Assessment forms are submitted via printed forms by students, while assessments by supervisors and panels are done by entering marks using the Google Forms function. While the process of calculating the overall score to produce evaluation results is also done manually (hand count) which leads to the risk of calculation errors. Thus, this web-based pre-viva assessment system has been developed to improve the existing process especially the application process, evaluation, and calculation of marks. The system also simplifies the process of data entry, information retrieval and management, as well as better analysis. System development follows the prototype model. The users of the system consist of administrator, supervisors, examiners, and students. The system implementation uses MySQL software via XAMPP, PHP, and Visual Studio Code. The developed system will be fully functional at the end of the project and provide faculty (FSKTM) with a web-based online platform to improve management quality from manual methods to centralized electronic methods that support pre-viva process efficiency.

Keywords: Pre-viva evaluation system, Web-based information system, Web technologies, Prototyping model.

1. Introduction

Master's and Doctor of Philosophy by research is a graduate program based entirely on research work, which culminates in the writing and submission of a thesis for oral examination, or viva-voce. Viva-voce is a long-established part of the examination process for a research degree [1] which normally is an oral test that focused discussion giving postgraduate student the opportunity to defend the thesis with the panel of educational specialists [2]. It is the University's responsibilities to develop and manage a robust assessment system which is secure, of quality and is subjected to review and improvement periodically [3]. Pre-viva assessment system in FSKTM UTHM is a system designed to mock the viva-voce and the main motivation of the pre-viva assessment system is to assure that the postgraduate

candidate's thesis is significant which in compliance with thesis writing format and holds befitting academic merit.

The case study for the pre-viva assessment system is conducted in Faculty of Computer Science and Information Technology (FSKTM) postgraduate unit of Universiti Tun Hussein Onn Malaysia (UTHM). The administrative staff of the Faculty Postgraduate Unit handles the pre-viva assessment process. Students, supervisors, and two internal examiners all participated in this procedure. Currently, the pre-viva assessment method is done manually, with all pre-viva assessment processes submitted in printed form. The thesis draft will be distributed for examination to the supervisor and two examiners. Marks will be awarded based on assessment items based on thesis draft and pre-viva oral sessions. The supervisor and two examiners will make suggestions and comments for improvement for students to make corrections.

Currently, the pre-viva evaluation process is done entirely manually, and is beginning to be done hybrid, partly using printed evaluation forms, and using the online form function (google forms). This results in assessment documents and data being stored centrally. Meanwhile, the process of calculating the overall score to produce the evaluation results is also done manually (hand counting) which leads to the risk of calculation errors. From this point of view, there are some limitations to the existing pre-viva evaluation procedures regarding the submission of thesis files, providing feedback on the thesis and marking on the thesis. This is complicated because administrators, supervisors, examiners, and students need to manage information across multiple platforms, and part of the process needs to be done manually. In addition, a lot of time is required to complete relevant reports and analyzes which are seen to reduce the efficiency of the process itself especially in terms of its data management. Therefore, a web-based pre-viva assessment system is proposed to eliminate the pre-viva assessment procedure performed manually at present.

With the proposed system, administrators are responsible for managing student, supervisor and examiner information and assigning supervisors to each student through an administrative module. In addition, postgraduate students can apply for pre-viva assessment through this system by providing a proposal date and uploading a thesis draft and plagiarism report. Once the student submits the application, the supervisor receives the application including the student details and the thesis draft uploaded by the student. Supervisors can accept or reject pre-viva applications using the provided function in the system with an acceptable reason. Supervisors and examiners can review the thesis uploaded by the student and then provide amendments and marks to the thesis through a web-based system instead of filling out the evaluation form manually. Once the assessment is completed, the system produces a full decision result i.e., either pass or fail, automatically. This eases the struggles to calculate marks manually and to reduce calculation errors. In short, the proposed web-based pre-viva assessment system can help faculty to manage assessment data and to implement the assessment process more efficiently.

This paper is organized into five sections. The first part is an introduction that describes the context of the project, namely application management and pre-viva evaluation in FSKTM. The second section describes the related work. In the third section, the methodology is explained. The results and discussion are described in the fourth section. In the last section, a conclusion is given.

2. Related Work

The pre-viva assessment process is currently handled by administrative staff in the Faculty Postgraduate Unit. The parties involved in this process are students, supervisors, two internal examiners. At present, the pre-viva assessment system is done manually where all the processes for performing the pre-viva assessment are submitted in a printed form format. First, postgraduate students are required to complete a thesis based on the agreed study and submit three copies of the thesis draft hardcopy, plagiarism report and English proofread report to the faculty. After that, the thesis is distributed to the supervisor and two

examiners for evaluation. During the oral session of pre-viva, marks will be given based on the evaluation items. Students need to achieve a score of at least 80 marks to pass. Students need to repeat the pre-viva session if they do not reach the passing level.

The supervisor and two examiners will provide necessary suggestion for amendments to the thesis such as title, abstract, introduction, problem statement, research objectives, literature review, research design and the analysis of the research outcome. Postgraduate students are required to make corrections based on the amendments provided by the supervisor and two examiners. Students need to resubmit the thesis correction to the faculty for examination and verification by the supervisor and examiners. Once the supervisor and examiners approve the final thesis correction, the postgraduate student is required to return the draft of the corrected thesis to the Postgraduate Unit, together with the VIVA examination application. The draft of this thesis will be brought to the meeting of the Faculty Postgraduate Studies Committee (JKPSF) for further processing before being forwarded to the University Graduate School for VIVA examination purpose.

Pre-viva assessment score calculation form which is done manually in the paper printed form. Scores are calculated with Equation (1) to Equation (3).

$$M_{pt} = 0.5(M_{pt_1} + M_{pt_2}) \quad (1)$$

$$M_{pp} = 0.5(M_{pp_1} + M_{pp_2}) \quad (2)$$

$$M = 0.6M_s + 0.2M_{pt} + 0.2M_{pp} \quad (3)$$

The thesis evaluation score of the supervisor is M_s , while the thesis evaluation score of the two panels is M_{pt} . M_{pp} is the score from the two panels for student's presentation.

The student's pre-viva marks and results will be determined by the total marks from the supervisor and these two assessors. Finally, this evaluation and scoring process comprises a well-thought-out procedure and technique. However, it is still implemented using paper and printed forms, and it is kept in physical files. Hand calculations are used to perform the calculation procedure. It has been discovered that the existing technique of process implementation is prone to calculation errors, the use of paper forms, and a lot of physical space, and the data is not easily available at an immediate rate. Consequently, a web-based system and database have been recommended as a method of increasing process efficiency.

A web-based information system is an information system that employs Internet web technologies to deliver information and services to consumers. Information systems are a combination of information technology and the actions of people who use it to aid in operations, management, and decision-making [4]. The information system stores documents, review history, communication records, and operational data. According to the five-component model of an information system, software is the most important component of an information system. Corporate organizations use this web-based information system to expand and run their operations globally [5]. The web-based technology also plays an important role in education field information system. It allows users to fully access the system and obtain any information such as learning materials, assignment tasks, grades at anytime and anywhere, if there is Internet and smart gadget devices. Web-based information systems have grown considerably in recent years as technology has advanced and there are many advantages of cross-platform interoperability, management, numerous concurrent users, and lower expenses in development. Therefore, the most suitable platform for administering e-learning processes where courses and learning items are given via the internet is undoubtedly a web-based information system [6].

The project requires an online information system, to collect pre-viva application details, task assignment, scoring, notifications, and reporting, The task is an essential part of the pre-viva application procedure. This new project's goal is to collect and disseminate information among stakeholders while

also increasing productivity. To make reporting easier, data should be collected and managed centrally. The postgraduate unit can process the data collected to provide helpful information for the pre-viva assessment procedure. Taking into mind the benefits of information system and Internet technologies [7], the project can make use of them to boost productivity [8]. It is expected that by incorporating elements of information systems, information technology, and the Internet into the construction of this new system, it will increase the efficiency of the implementation of the pre-viva application and evaluation process.

Study has been conducted to compare between current similar system and the proposed system, and the findings are briefly summarized in **Table 1**. This web-based pre-viva assessment system proposes some similar modules from the compared systems, but the project system focuses mainly on assessment system.

Table 1: Comparison analysis among similar systems

System/Features	Edmodo	Google Classroom	AUTHOR UTHM	FSKTM Pre-Viva Assessment System
Module	Learning management and assessment	Learning management and assessment	Learning management and assessment	Management and assessment
Login	Email and password	Email and password	Matric number or staff id and password	Matric number or staff id and password
Upload document	Lecturer and student both can upload document	Lecturer and student both can upload document	Lecturer and student both can upload document	Lecturer and student both can upload document
Administrator Panel	Not available	Administrator used to manage system	Not available	Administrator used to manage system
User	Educator, student, and parent	Administrator, educator, and student	Academic staff and student	Administrator, academic staff, and student
Security	Provide authorization access	Provide authorization access	Provide authorization access	Provide authorization access
Platform	Responsive Web-based and	Responsive Web-based & Mobile application	Responsive Web-based	Responsive Web-based

Mobile
application

3. Methodology

Methodology is the framework of the entire process of information system development through a multi-step process. These range from initial needs research to SDLC, including analysis, design, implementation, and maintenance.

3.1 Prototyping Model

The Prototype model is one of the software development life cycle models in which a prototype is built with minimal requirements [9]. Prototyping ensures that the end users constantly work with the system and provide feedback which is incorporated in the prototype to result in a useable system [10]. **Table 2** lists each phase with its own assignment and output that need to be produced during the entire project development. Besides that, the output had been completed within the specific days that have been given [11].

Table 2: Software development activities

Phase	Activity	Deliverable
Requirement gathering and analysis	Identify the stakeholder, identify the problem, and study what system feature should have in pre-viva assessment system	<ul style="list-style-type: none"> • The proposal • System Requirement • Data Flow Diagram • Entity Relationship Diagram
Quick Design	Decide the software and hardware requirement, design user interfaces	<ul style="list-style-type: none"> • System Architecture • Interfaces of system • Database design
Build Prototype	Develop the system based on the quick design	<ul style="list-style-type: none"> • Prototype of Pre-viva assessment system
Evaluation	Demonstrate the functional and non-functional features of the system of Prototype I	<ul style="list-style-type: none"> • Feedbacks and suggestions report on Prototype
Refining Prototype	Develop another prototype based on the feedbacks and suggestion from evaluation phase	<ul style="list-style-type: none"> • Latest Prototype of Pre-viva assessment system
Evaluation	Demonstrate the functional and non-functional features of the system of the latest Prototype	<ul style="list-style-type: none"> • Feedbacks and suggestions report on Prototype • Approval of final system development
Implementation and Maintenance	Develop, test, and deploy the final system based on the approved final prototype	<ul style="list-style-type: none"> • System software • Report • User manual

3.1.1 Requirement Gathering and Analysis Phase

All possible requirements information for the proposed system is captured and documented in a requirement specification document. During this phase, discussions and interviews were conducted with

FSKTM postgraduate officers to obtain information on the pre-viva assessment process. All relevant forms used in greeting the existing assessment process are now also collected and reviewed. Google Meet is a web-based meeting platform for conducting interviews with stakeholders. The purpose of this interview is to learn about the current process, acquire system needs, and other relevant information. This information is critical in understanding the project's broad context, difficulties faced, and potential solutions, which are subsequently presented in a project proposal document.

Table 3: System functional modules of pre-viva assessment system

Module	Function	User
Login module	<ul style="list-style-type: none"> • Authentication of user who can log in and use the system. • A user who has the valid id and password can only login to their account respectively. 	<ul style="list-style-type: none"> • Administrator • Supervisor • Examiner • Postgraduate student
Profile management module	<ul style="list-style-type: none"> • Database management (Create, read, update, and delete information) 	<ul style="list-style-type: none"> • Administrator
Pre-viva application module	<ul style="list-style-type: none"> • Student applies for pre-viva • Student upload thesis draft, plagiarism report, and proofread receipt • Supervisor verifies the draft thesis is ready for evaluation • Supervisor nominates 2 internal examiners and pre-viva date. 	<ul style="list-style-type: none"> • Postgraduate student • Supervisor • Administrator
Assessment module	<ul style="list-style-type: none"> • Supervisor and examiners enter the mark • System generates the result • Supervisor and examiners give recommendation and comments and upload a feedbacks version of thesis draft through the system. • System generates full comments report for correction 	<ul style="list-style-type: none"> • Supervisor • Examiner
Thesis correction verification module	<ul style="list-style-type: none"> • Student upload correction version of thesis draft • Student upload correction report for committee verification • Examiners check the correction on the thesis and correction report • Examiners verify the correction had been made • System generates thesis correction report (verified) 	<ul style="list-style-type: none"> • Examiner
Report module	<ul style="list-style-type: none"> • Generate the result from assessment (pass/fail and marks) • Generate analysis (number of students pass/fail) 	<ul style="list-style-type: none"> • Postgraduate student • Administrator

All information collected is then analyzed. Flowcharts are generated to illustrate the existing process, and the proposed system. Requirement specification document are prepared based on the

collected information. The proposal of the pre-viva assessment system is written. Project requirement, system requirement, data flow diagram and entity relationship diagram are also produced based on the requirements. In this phase, problems of the current system will be analyzed and then study what system features and what kind of stakeholder should include in the pre-viva assessment system. **Table 3** summarizes the functional modules provided in the system.

Functional requirements define the function of the developed system, while function is described as specific behavior that convert input to output. **Table 4** shows the functional requirements of the proposed system. Besides that, non-functional requirements define the criteria that is used to judge the operation of a system, rather that the specific behavior or function of the system. **Table 5** shows the non-functional requirements of the developed system.

Table 4: Functional requirements

No	Module	Description
1	Login module	<ul style="list-style-type: none"> • The system should allow user to login into the system using registered user id and password. • The system should only allow a user to log in as a user with a valid user id and password. • The system should alert the user for any invalid user credentials. • The system should redirect user to that respective main homepage upon successful login.
2	Profile Management module	<ul style="list-style-type: none"> • The system should allow administrator to add, edit, and delete user's information. • The system should allow administrator to pre-set a supervisor to each student. • The system should alert administrator if there are invalid input.
3	Pre-viva application module	<ul style="list-style-type: none"> • The system should allow students to upload the thesis draft, plagiarism report, proofread receipt during pre-viva application process. • The system should allow supervisor to review the pre-viva application details. • The system should allow supervisor to nominate maximum two internal examiners for the student. • The system should allow supervisor to approve or reject the pre-viva application. • The system should allow administrator to verify the completeness of pre-viva application. • The system should allow administrator to approve or reject the pre-viva application.
4.	Assessment module	<ul style="list-style-type: none"> • The system should allow supervisor and examiners to evaluate the thesis draft uploaded by the student once the administrator approved the application. • The system should allow supervisor and examiners to enter comments or recommendations during assessment process. • The system should calculate the overall assessment mark and store into database.

5.	Thesis correction module	<ul style="list-style-type: none"> The system should allow student to upload thesis correction after he or she passed the thesis evaluation. The system should allow examiners to verify the thesis correction either approve or reject.
6.	Report module	<ul style="list-style-type: none"> The system should generate a result of pass or fail to the student and administrator. The system should generate overall comments or recommendations upload by the supervisor and examiners to the student. The system should generate analysis of number of student pass or fail the pre-viva assessment.

Table 5: Non-functional requirements

No	Requirements	Description
1.	Performance	<ul style="list-style-type: none"> The system should be always usable
2.	Operational	<ul style="list-style-type: none"> The loading time required for a website is no more than 1 minute
3.	Security	<ul style="list-style-type: none"> The system should be user friendly
4.	Cultural and political	<ul style="list-style-type: none"> The system should be able to work on any web browser

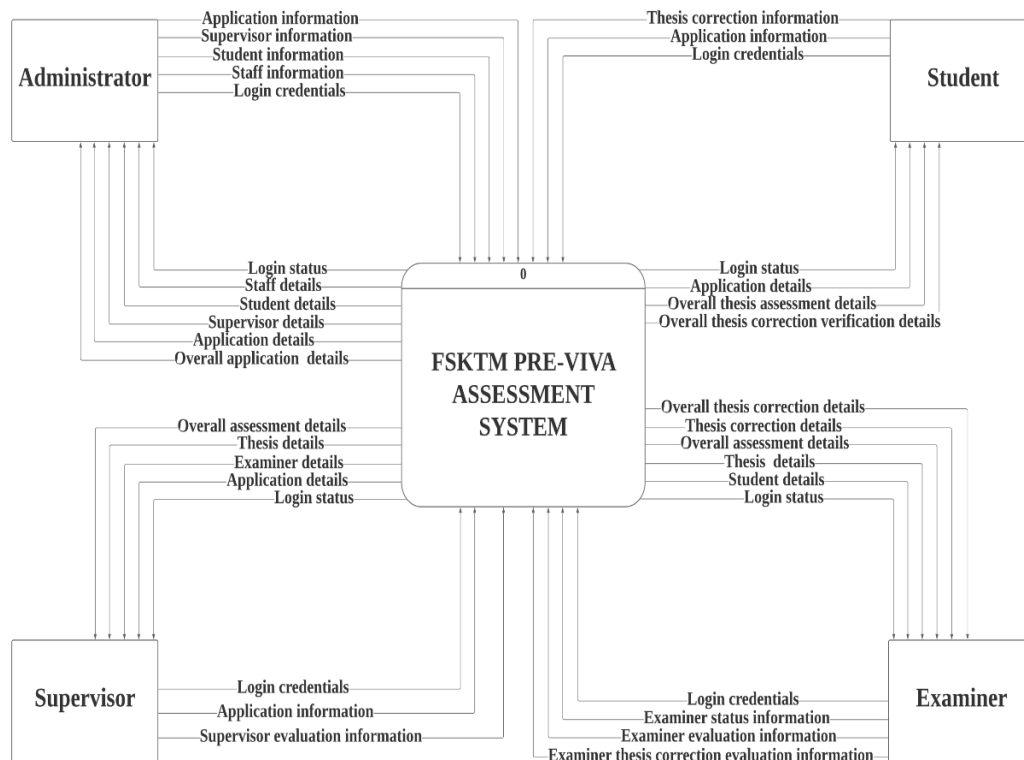


Figure 1: Context diagram

System analysis provides the following specification. The analysis is conducted using structured approach. Context diagrams present the overview of interaction between the system and its user.

Context diagram also show the input and output to and from its user and system. **Figure 1** shows the context diagram and **Figure 2** shows the Level 0 diagram of the developed system.

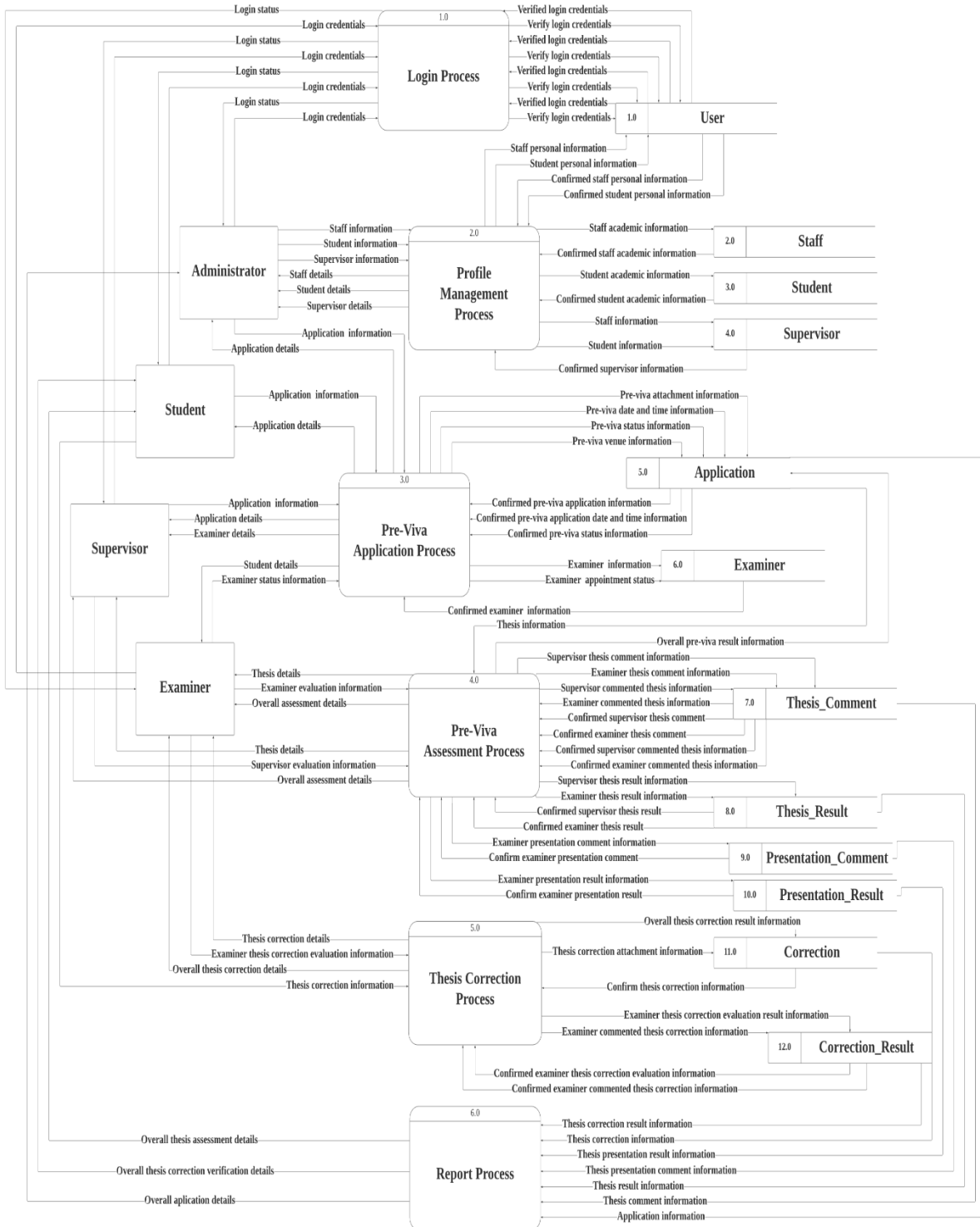


Figure 2: Level 0 Diagram

An Entity Relationship Diagram (ERD) is used to describe and explain the components involved in designing the database where the information of each table can be linked together. ERD aims to ease the development of designing a system for managing the information of the system. **Figure 3** shows the ERD for the FSKTM pre-viva assessment system and has ten entities which are user, staff, student, supervisor, examiner, application, result, comment, correction, and correction result.

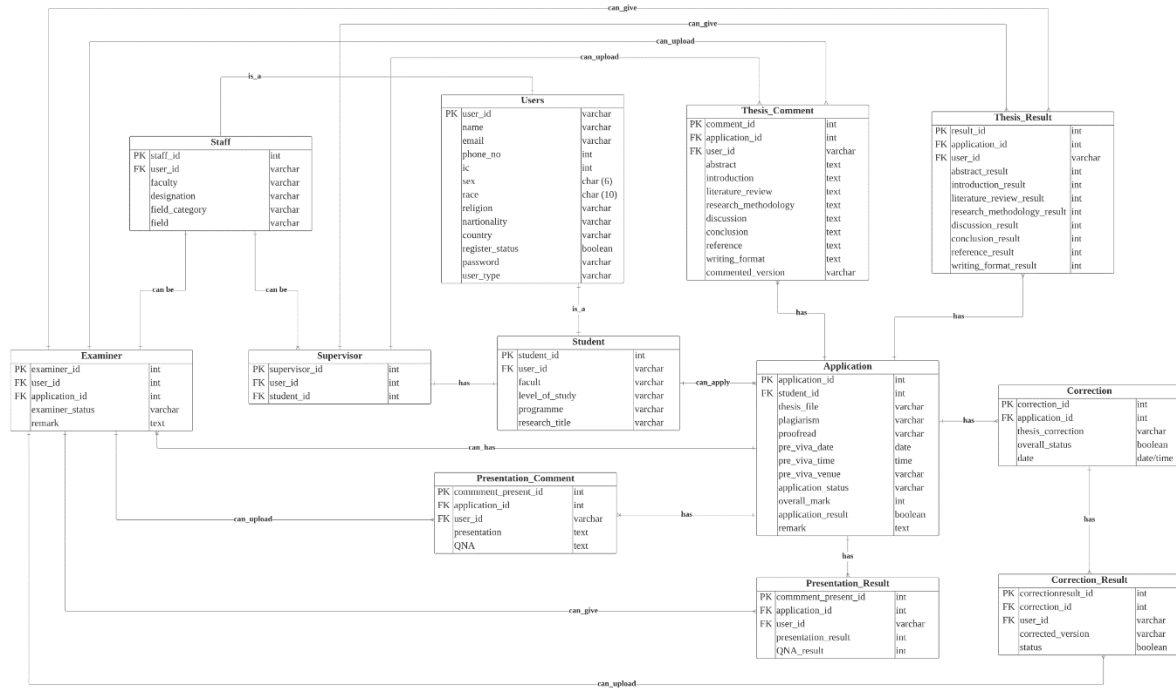


Figure 3: Entity relationship diagram

A flowchart is a visual representation of a programmer's thoughts or solutions to current problems [12]. The hardest processes are understood easily by a flowchart [13]. **Figure 4** shows the flowchart of the FSKTM pre-viva assessment system.

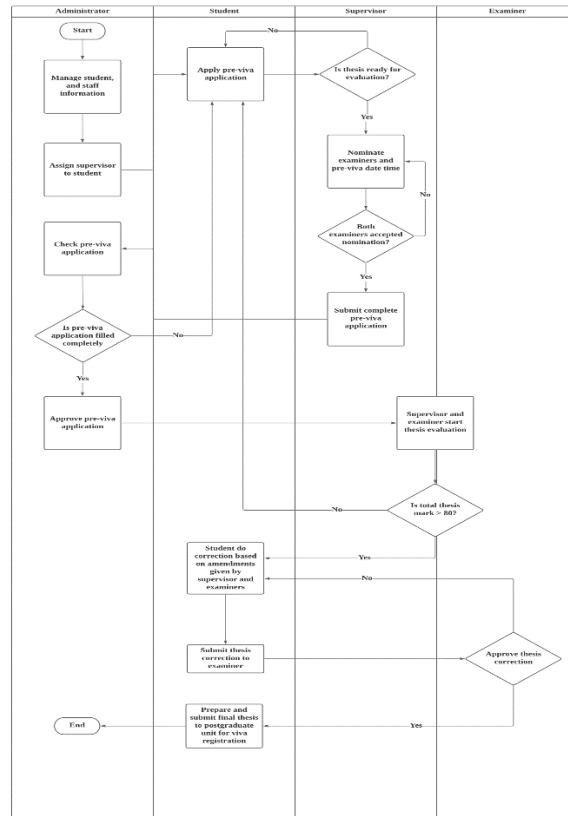


Figure 4: System flowchart

3.1.2 Quick Design Phase

Quick design phase which to decide the software and hardware requirement and design the user interfaces and database. At the design phase, a simple of the system is created. However, it is not a complete design, but it gives a brief idea of the system to the user. The quick design phase used to help in developing the prototype. Microsoft Word is used to sketch the interfaces of the system. **Figure 5** shows the fundamental system architecture. In addition, the user interface of the system is shown in **Figure 6 to Figure 9**.

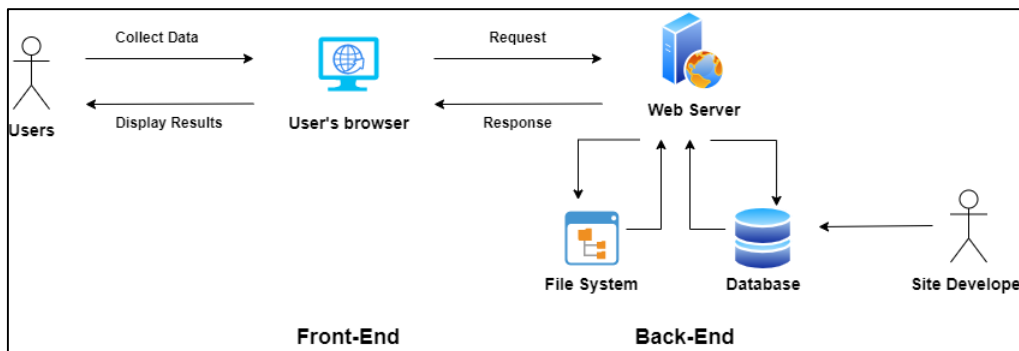


Figure 5: System architecture

The relational schema for database tables is listed as follows:

- i. Users (user_id, name, email, phone no, ic, sex, race, religion, nationality, country, register_status, password, user_type)

- ii. Staff (staff_id, user_id, faculty, designation, field_category, field)
- iii. Student (student_id, user_id, faculty, level_of_study, programme, research_title)
- iv. Supervisor (supervisor_id, user_id, student_id)
- v. Examiner (examiner_id, user_id, application_id, examiner_status, remark)
- vi. Application (application_id, student_id, thesis_file, plagiarism, proofread, pre_viva_date, pre_viva_time, pre_viva_venue, application_status, overall_mark, application_result, remark)
- vii. Thesis_Comment (comment_id, application_id, user_id, abstract, introduction, literature_review, research_methodology, discussion, conclusion, reference, writing_format, commented_version)
- viii. Thesis_Result (result_id, application_id, user_id, abstract_result, introduction_result, literature_review_result, research_methodology_result, discussion_result, conclusion_result, reference_result, writing_format_result)
- ix. Presentation_Comment (present comment id, application_id, user_id, presentation, QNA)
- x. Presentation_Result (present result id, application_id, user_id, presentation_result, QNA_result)
- xi. Correction (correction_id, application_id, thesis_correction, overall_status, date)
- xii. Correction_Result (correctionresult_id, correction_id, user_id, corrected_version, status)

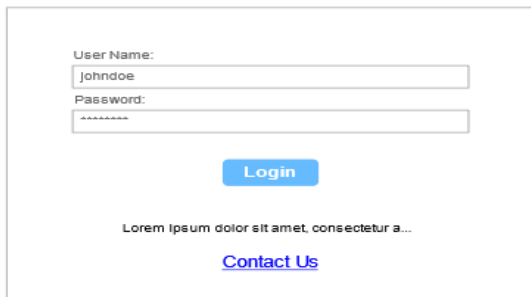


Figure 6: Login page

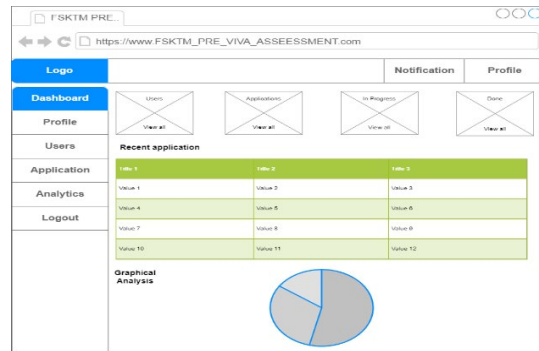


Figure 7: Administrator's Dashboard

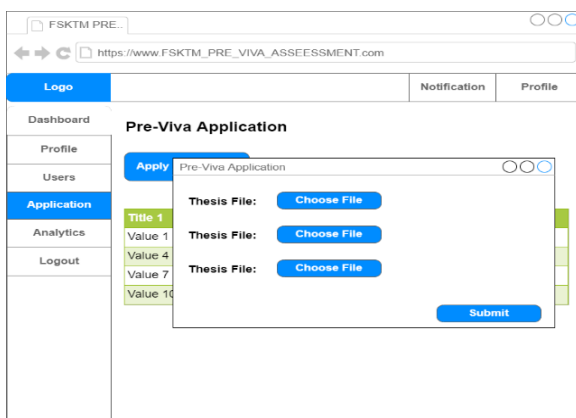


Figure 8: Pre-viva application page

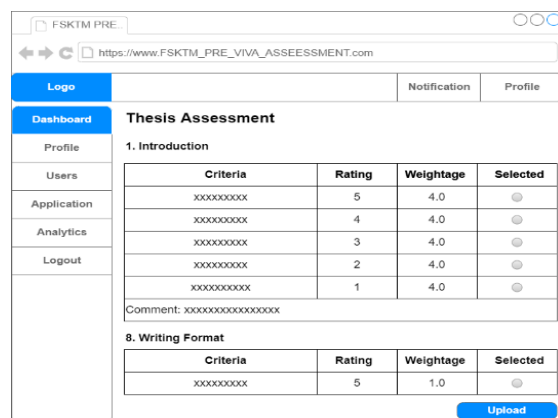


Figure 9: Assessment page

3.1.3 Build Prototype Phase

In the build prototype phase where an actual prototype of pre-viva assessment system is coded based on the information gathered from quick design. It is a small working model of the pre-viva assessment system. Visual Studio Code is used to code the interfaces of the system using HTML, JavaScript and CSS programming language and phpMyAdmin is used to create and manage the database of the system.

3.1.4 User Evaluation Phase

In user evaluation phase, the small working model of the pre-viva assessment system is presented to the administrator, academic staff (supervisor and examiner), and student for an initial evaluation which is the feedbacks report on the prototype. It helps to find out the strength as well as the weakness of the working model of the pre-viva assessment system and necessary comments or suggestions are collected from the client for improvement. Google Meet is used to demonstrate the functional and non-functional features of the prototype and then collect the feedbacks and suggestions report on the prototype.

3.1.5 Refining Prototype Phase

In refining prototype phase, the small working model of the pre-viva assessment system is modified based on the feedbacks and suggestions from the evaluation phase. Once the prototype is done modifying, the modified prototype will undergo evaluation phase again where the prototype is present again to the client environment and collect feedbacks until the client satisfied with the prototype and approve for final system development.

3.1.6 Implementation and Maintenance Phase

Implementation and maintenance phase would proceed if the client approved the final prototype for system development. The system will be developed based on the latest prototype and it will be fully tested and deployed to production. The system will undergo routine maintenance to minimize downtime and prevent large-scale failures.

4. Result and Analysis

This section describes the system implementation process and the system testing for the FSKTM Pre-Viva Assessment System.

4.1 System Implementation

System implementation involves all the processes such as source code of the system development.

4.1.1 Login module

Figure 10 shows the interface for the login module and users are asked to enter the user credentials to log into the system. There are basically 3 user types, which are administrator, academic staff, and student. If the user credentials are correct, user will be redirected to corresponding homepage.

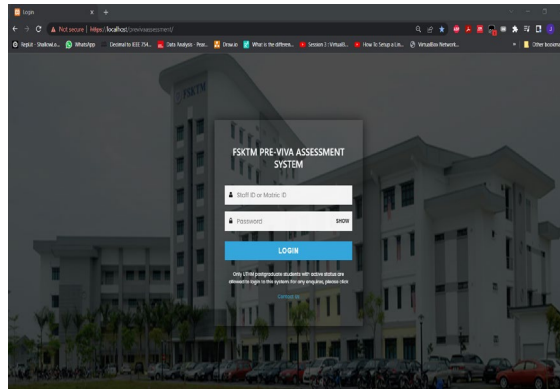


Figure 10: Login page

4.1.2 Profile management module

Figures 12, 13, and 14 depict the user interface of profile management module for the administrator, academic staff, and student respectively. Basically, the super administrator can add a sub-administrator to help to manage the system, and the administrator can add new academic staff, and student information into database.

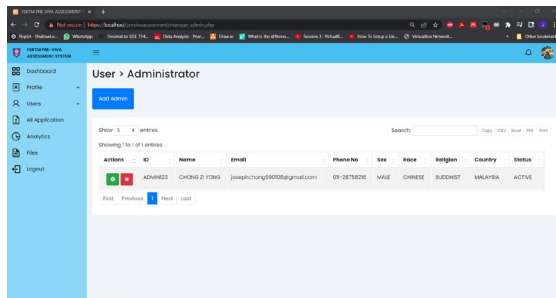


Figure 12: Profile management page (Admin)

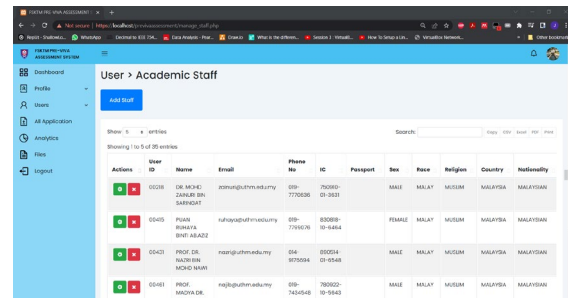


Figure 13: Profile management page (Staff)

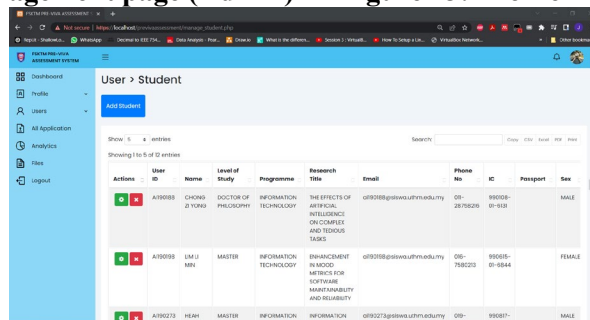


Figure 14: Profile management page (Student)

4.1.3 Pre-viva application module

Figures 15 to 18 show the user interfaces for the pre-viva application procedures for the student, supervisor, examiner, and administrator respectively. To apply for the pre-viva exam, the student will upload the thesis file, the thesis plagiarism report, and the proofread receipt. Aside from that, the supervisor will conduct a brief check of the thesis to verify that it is ready for evaluation by scheduling a pre-viva date and time and appointing no more than two assessors. The examiner has the option of accepting or declining the examiner invitation. After all examiners have accepted the invitation, the administrator can check the application information for completeness and approve the application by selecting the pre-viva venue platform and venue.

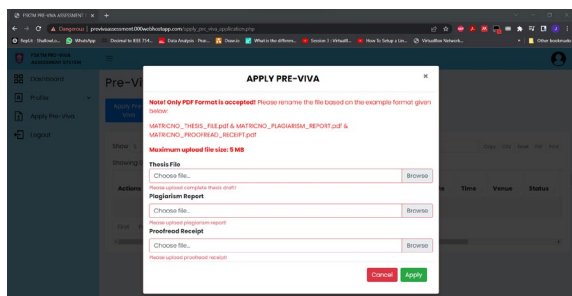


Figure 15: Pre-viva application page (Student)

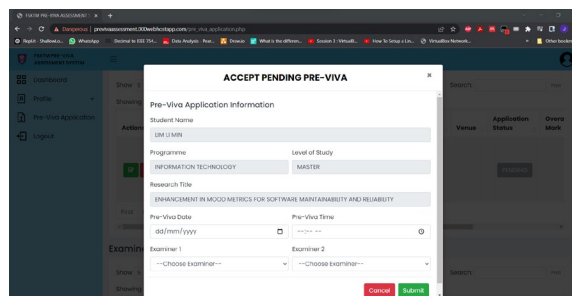


Figure 16: Pre-viva application page (Supervisor)

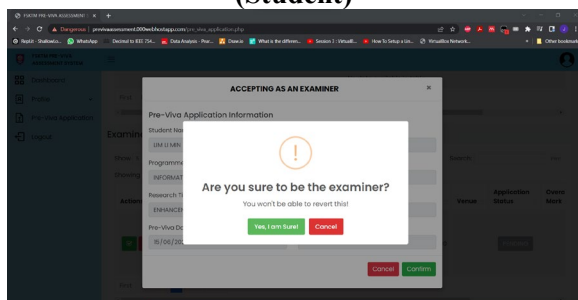


Figure 17: Pre-viva application page (Examiner)

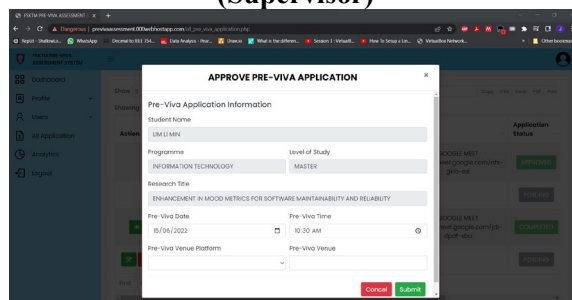


Figure 18: Pre-viva application page (Administrator)

4.1.4 Pre-viva assessment module

The supervisor and examiner user interfaces for the pre-viva assessment procedures are shown in **Figures 19** and **20**. The pre-viva assessment processes are divided into two sections: thesis evaluation and presentation evaluation. The thesis can be evaluated by both the supervisor and the examiner, but only the examiner can evaluate the presentation. Students must get a minimum score of 80 out of 100 on the thesis evaluation before attending the thesis presentation. If either the supervisor's or examiner's thesis evaluations are less than 80, the student will automatically fail the pre-viva assessment and must apply for the evaluation again. The examiner can evaluate the student's presentation performance after the presentation date has passed. The total pre-viva result will be determined automatically once the presentation assessment procedure is done, and a pass or fail result will be generated for all users.

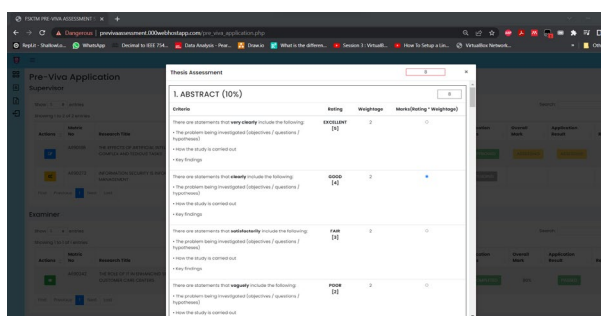


Figure 19: Thesis assessment user interface

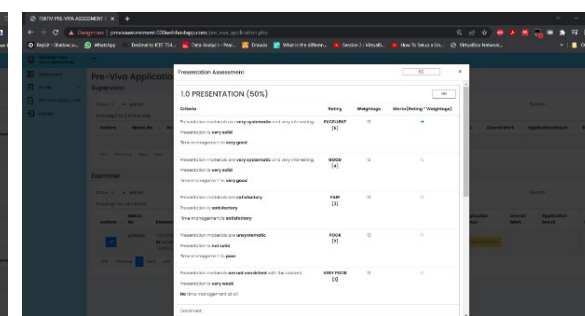


Figure 20: Presentation assessment user interface

4.1.5 Report module

The report module user interface is shown in **Figures 21** and **22**. **Figure 21** provides an overview to all four categories of users of the supervisor's and examiner's comments and marks given during the pre-viva assessment procedures. **Figure 322** shows the total pass and fail percentages for all pre-viva applications, allowing the administrator to analyse and refine the guidelines to assessors to help students improve their thesis writing and presentation abilities before taking the real viva examination.

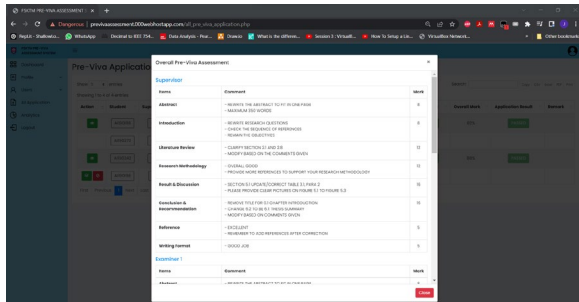


Figure 21: Pre-viva assessment report module

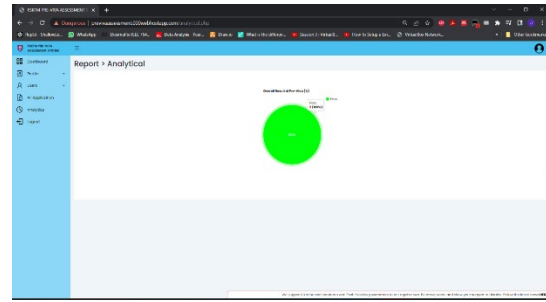


Figure 22: Pie chart report module

4.2 Testing

The testing step is used to determine whether the system's efficacy matches the system requirements set forth by the users. The goal of this testing is to guarantee that the system's components work properly and without mistakes. This test case includes the login module, profile management module, pre-viva application module, pre-viva assessment module, and report module, which are all accessible in the FSKTM Pre-Viva Assessment System. Users conducted several tests throughout the login procedure, as shown in **Table 6**.

Table 6: Test case of login module

No	Description	Expected	Actual	Result
TEST_100_001	Verify if the username is incorrect or account is inactive	The user should be alert with an error message	The error message displayed to user	Pass
TEST_100_002	Verify if the password is incorrect	The user should be alert with an error message	The error message displayed to user	Pass
TEST_100_003	Verify if the username is empty	The user should be alert with an error message	The error message displayed to user	Pass
TEST_100_004	Verify if the password is empty	The user should be alert with an error message	The error message displayed to user	Pass
TEST_100_005	Verify if both username and password are empty	The user should be alert with an error message	The error message displayed to user	Pass
TEST_100_006	Verify if the username and password are matched	The user should be redirect to the home page of the system	The user is redirected to the home page respectively	Pass

Table 7: Test case of profile management module

No	Description	Expected	Actual	Result
TEST_200_001	Verify whether the administrator has completed all mandatory fields on the add user form	The administrator should be alert by the validation error message	Data entry into the database is prohibited for administrators, and a validation error message is displayed.	Pass
TEST_200_002	Verify whether the user's information has inserted into database	An insert success message should notify the administrator and the data should be stored into database.	A successful insert message is displayed, and data has been saved to the database.	Pass
TEST_200_003	Verify whether the administrator has completed all mandatory fields on the edit user form	The administrator should be alert by the validation error message	Data entry into the database is prohibited for administrators, and a validation error message is displayed.	Pass
TEST_200_004	Verify whether the user's information has updated into database	The administrator should be notified with an insert success message, and the database should be updated.	A successful update message is displayed, and data has been saved to the database.	Pass
TEST_200_005	Verify whether the administrator deleted the correct user when the delete button is clicked	The administrator should be alert with a modal for confirmation to delete the user	The user's data is deleted from the database when the confirm button is clicked, and an administrator-approved modal is displayed.	Pass

The collection of test cases for the profile management module is shown in **Table 7**. All the test scenarios yielded the desired result. The goal of this test is to ensure that the create, read, update, and delete operations are running properly, and that validation is in place to prevent empty or invalid data from being put in the database.

Table 8: Test case of pre-viva application module

No	Description	Expected	Actual	Result
TEST_300_001	Verify that the documents uploaded by the student are correctly placed in the requested path and that the path name for each document and the application information have been entered into the database	The documents submitted by the student should be placed in the relevant folder, and the folder path should be kept in the database	The documents are placed in the correct folder path and the application information as well as the folder path are kept into database	Pass
TEST_300_002	Verify that the documents uploaded by the student are not more than 5MB in total	If the total size of documents exceeded 5MB, an error message will be displayed	If the file size exceeded 6 MB, no error message displayed, and error page is shown	Fail
TEST_300_003	Verify that the supervisor selected	The application information	The application information is updated,	Pass

	pre-viva date, time, and examiners information are stored into the database	should be updated into tbl_application table and examiner information should be inserted into tbl_examiner table	and a new record of the examiner information is inserted into database	
TEST_300_004	Verify that the supervisor provided reason of rejecting the pre-viva application information are stored into the database	The application information should be updated into tbl_application table	The application information is updated, and an update successful alert message will be displayed	Pass
TEST_300_005	Verify that the examiner information is saved in the database once the invitation has been accepted	The status of examiner should be updated into tbl_examiner table	The status of examiner is updated, and a accept successful alert message will be displayed	Pass
TEST_300_006	Verify that the examiner provided reason of rejecting the examiner invitation are stored into the database	The reason and status information should be updated into tbl_examiner table	The examiner information is updated, and a reject successful alert message will be displayed	Pass

Table 8: (cont.)

No	Description	Expected	Actual	Result
TEST_300_007	Verify the application information have been stored into database once administrator approve the pending pre-viva application	The application information should be updated, and the assessor side should appear a button for thesis assessment	The application information updated into database, and a "Thesis Assessment" button appeared	Pass
TEST_300_008	Verify the application information have been stored into database once administrator reject the pending pre-viva application	The application should be updated, and the student side should appear again "Apply Pre-Viva" button	The application information updated into database, and the "Apply Pre-Viva" button appeared again at student part.	Pass

Table 9: Test case of pre-viva assessment module

No	Description	Expected	Actual	Result
TEST_400_001	Verify that the thesis assessment modal is pop out and the thesis assessment	A thesis assessment modal should display after the "Thesis Assessment" button is clicked, and the	A thesis assessment modal displayed, and the thesis information is inserted into database	Pass

	information are stored into database	thesis information should be inserted into database		
TEST_400_002	Verify that the commented thesis document uploaded by the assessors are not more than 5MB in total	If the total size of documents exceeded 5MB, an error message will be displayed	If the file size exceeded 6 MB, no error message displayed, and error page is shown	Fail
TEST_400_003	Verify that the application status will be updated once all the assessors completed the thesis evaluation process.	The application status should be updated into tbl_application table to indicate pass or fail of the thesis assessment	The application status updated into tbl_application table once all the assessors completed thesis assessment	Pass
TEST_400_004	Verify that the presentation assessment modal is pop out and the presentation assessment information are stored into database	A presentation assessment modal should display after the "Presentation Assessment" button is clicked, and the presentation information should be inserted into database	A presentation evaluation modal is shown, and the database is updated with the presentation data.	Pass
TEST_400_005	Verify that the application status and application result will be updated once all the examiners completed the presentation evaluation process.	The application status and application result should be updated into tbl_application table to indicate pass or fail the of the overall pre-viva assessment	Once all examiners had completed their presentation evaluations and a "View" button had appeared in the action column of the table, the application status and application result were updated into the tbl_application table.	Pass

Table 8 shows the total pass and fail of the test cases performed in the pre-viva application module. One of the test cases failed because the 000WebHost only allowed a maximum upload file size of 5MB, and if the student's file surpassed this limit, the 000WebHost would hang and display an error page. Other than that, another 7 test cases are functioning as desired.

Table 9 shows the total pass and fail of the test cases performed in the pre-viva assessment module. One of the test cases failed because the 000WebHost only allowed a maximum upload file size of 5MB, and if the assessor's file surpassed this limit, the 000WebHost would hang and display an error page. Aside from that, additional four test scenarios are running as expected, ensuring that legitimate data and automated calculations during the pre-viva assessment procedures are working well.

Table 10: Test case of report module

No	Description	Expected	Actual	Result
TEST_500_001	Verify that a "View Overall" appeared after examiners finished the presentation evaluation	Administrator, supervisor, examiners, and student should be able to click the "View Overall" button to review the pre-viva assessment information	Administrator, supervisor, examiners, and student able to click the "View Overall" button to review the pre-viva assessment information	Pass
TEST_500_002	Verify that a pie chart appeared after administrator clicked	Administrator should be able to see a pie chart that display an	Administrator able to see a pie chart that display an overall pre-viva	Pass

the “Analytical” at the sidebar	overall pre-viva application in term of pass and fail	application in term of pass and fail
---------------------------------	---	--------------------------------------

Table 10 shows the total pass and fail of the test cases performed in the report module. The goal of this testing is to check if the administrator, supervisor, examiners, and student all can overview the pre-viva assessment result. In addition, a pie chart should be displayed to the administrator when “Analytics” button at the sidebar is clicked to illustrate an overall pass and fail percentage of pre-viva applications.

5. Conclusion

In conclusion, a web-based pre-viva assessment system with functional modules that support the pre-viva assessment process has been successfully developed. The web-based pre-viva assessment system is a helpful informational tool that enables supervisors and examiners to impart helpful knowledge to students so they can improve before taking the actual viva examination. The newly developed pre-viva assessment system can significantly improve the efficiency of the procedure when compared to the implementation of current hybrid-manual system. For the future development of this Pre-Viva Assessment System, there are several recommendations. The first suggestion is to enhance the system by giving push notifications to every user. The second recommendation is to incorporate a mock thesis rectification module into the system. This would enable students to adjust their theses and obtain examiner approval before moving on to the actual viva.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support and encouragement.

References

- [1] Jackson, Carolyn, and Penny Tinkler. "Back to basics: a consideration of the purposes of the PhD viva." *Assessment & Evaluation in Higher Education* 26, no. 4 (2001): 355-366.
- [2] Davis, Geraldine, and Hilary Engward. "In defence of the viva voce: Eighteen candidates' voices." *Nurse education today* 65 (2018): 30-35.
- [3] Watts, Jacqueline H. "Preparing doctoral candidates for the viva: issues for students and supervisors." *Journal of Further and Higher Education* 36, no. 3 (2012): 371-381.
- [4] Davis, G. B. (2000). *Information systems conceptual foundations: looking backward and forward*. In *Organizational and social perspectives on information technology* (pp. 61-82). Springer, Boston, MA.
- [5] Bellaaj, Moez. "Web-based information systems success: testing the antecedents of online customer satisfaction." *International Journal of Internet and Enterprise Management* 8, no. 2 (2013): 129-145.
- [6] Angelini, Anna, Enrica Gentile, Paola Plantamura, and Vito Leonardo Plantamura. "Web Information System for e-Learning." *International Academy of Sciences: ENFORMATIKA* 8 (2005): 209-212.
- [7] C. Wankel, *Cutting Edge Social Media Approaches to Business Education*, Information Age Publishing, 2010, pp. 1-5.

- [8] Florea, Adrian, Arpad Gellert, Traian Anghel, Delilah Florea, and Samuel von Brukenthal. "Enhanced Learning and Educational Management through Online Collaborative Technologies." *J. Digit. Inf. Manag.* 9, no. 1 (2011): 33-42.
- [9] Munassar, Nabil Mohammed Ali, and A. Govardhan. "A comparison between five models of software engineering." *International Journal of Computer Science Issues (IJCSI)* 7, no. 5 (2010): 94.
- [10] Sabale, Rajendra Ganpatrao, and A. R. Dani. "Comparative study of prototype model for software engineering with system development life cycle." *IOSR Journal of Engineering* 2, no. 7 (2012): 21-24.
- [11] Braude, Eric J., and Michael E. Bernstein. *Software engineering: modern approaches*. Waveland Press, 2016.
- [12] Charntaweekhun, K., & Wangsiripitak, S. (2006). Visual programming using flowchart. In *2006 International Symposium on Communications and Information Technologies* (pp. 1062-1065). IEEE.
- [13] Roy, G. G., Kelso, J., & Standing, C. (1998). Towards a visual programming environment for software development. In *Proceedings. 1998 International Conference Software Engineering: Education and Practice* (Cat. No. 98EX220) (pp. 381-388). IEEE.