APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 5 No. 2 (2024) 273-290
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Enhancing FSKTM Pre-Viva Management Through Online System

Derrick Tee Yee Meng[1], Hairulnizam Mahdin[2]*

[1,2] *Fakulti Sains Komputer dan Teknologi Maklumat,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: hairuln@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2024.05.02.015

Article Info

Received: 13 June 2024
Accepted: 28 September 2024
Available online: 15 December 2024

Keywords

Pre-viva management system, web-based, waterfall modal, 000webhost

Abstract

The pre-viva management system is a system that enables the postgraduate student to apply for pre-viva and for admin to manage pre-viva. The current pre-viva system in FSKTM is using internet tools and hand-writing of the application. This make the pre-viva application become no efficiency. Therefore, a web-based pre-viva management system will be proposed and developed to improve the process. The system development followed the waterfall methodology. The system will be implemented using XAMPP, PHP, Visual Studio Code, and 000webhost. At the end of the project, a fully functional web-basedonline system will be developed. Now the web-based system had been develope completely with user-friendly interface

1. Introduction

Viva voce has brought meaning as “living voce” in Latin, an oral examination where the postgraduate student must showcase their result expertise [1]. In some institutions, before the viva voce process, they conduct a pre-viva session. A pre-viva session is like a trial examination to ensure students are ready to face the actual viva session. In UTHM, it is a compulsory examination for master's and PhD students before attempting the final viva voce. Students must get at least 80% to pass the pre-viva session.

FSKTM used paper forms for the pre-viva process, making the process inefficient in managing the user's data. Students will submit the thesis draft, similar report, and the application form of pre-viva to the postgraduate unit in the faculty. The staff needs to arrange those documents according to the student's supervisor and the student's program. If there is any update, then the staff in the postgraduate unit need to search from those documents to find that student documentation. Therefore, this web-based pre-viva system improves the process of managing data and becomes more efficient

With this proposed system, all the manual processes will be eliminated and will be done by the system through the website. Administrators can manage the information of the student and supervisor through the administrator module provided in the system. Additionally, the administrator can also schedule the pre-viva date, venue, and platform for the student, and then the system will send that information to the supervisor, students, chairman, and examiner via email. Moreover, students can directly apply the pre-viva through the system by uploading the thesis draft and plagiarism report. Once the application form is submitted, the supervisor will receive the application form and the documents that are uploaded by the student. The

supervisor can choose to accept or reject the application form with a reason. Next, supervisors and examiners can also give the mark for the thesis draft and the performance of the student through the system with the provided form. Once finished, then the system will automatically calculate the overall mark of the student. Staff in the postgraduate unit did not need to calculate the mark manually and this reduced the risk of calculation error.

2. Related Work

Currently, the pre-viva process in FSKTM is handled by the administrator in the Faculty Postgraduate Unit. 5 parties are involved in the pre-viva process which are students, supervisors, examiner, chairman, and administrator. The pre-viva management system was done manually and all documents about the pre-viva process are submitted in paper form. First, students need to submit three thesis drafts followed by the plagiarism report and the thesis draft in hardcopy to the faculty. The thesis draft will be distributed to the supervisor and examiners for evaluation. During the pre-viva examination, the examiner and supervisor will give marks for the presentation section and the Q&A section via a Google form. Once the pre-viva examination is finished then the form will be submitted to the faculty to calculate the overall marks. Students are required to get at least 80 marks to pass the pre-viva examination. Otherwise, students need to repeat the pre-viva session until reach the passing level to sit for viva voce.

Additionally, the supervisor and the examiner will give their feedback on the elements in the thesis draft such as abstract, introduction, literature review, research methodology, discussion, conclusion, reference,and the writing format. Postgraduate students are required to make corrections according to the comments provided by the supervisor and examiner. Next, students need to resubmit the thesis correction to the faculty again for examination and verification by the supervisor and examiners. Once the thesis correction has been approved by the supervisor and examiner, the student is required to return the thesis draft to the Postgraduate Unit.

2.1 Approach and technology of system

The Pre-viva Management System was developed using a web-based approach which is an application that is accessed via HTTP. Additionally, a web-based application requires internet connectivity to deliver the service to the users. With the internet provision from the UTHM, the supervisor and the student can visit the system through the web browser. Web-based approach refers to a system that uses the Internet as a platform for managing all the processes in the system. There will be some technology chosen to be used in developing the system such as PHP programming, XAMPP, and Visual Studio Code. Additionally, the pre-viva management also been host by using 000webhost which is a free hosting service pervider and support phpMyadmin with MySQL.

2.2 Study of the existing system

The study on the existing system was carried out to make a comparison between a similar system and the proposed system. Table 1 shows the comparison between three similar systems and the proposal system.

Table 1 : Compatison analysis between similar system

System / Features	**Author**	**Google Classroom**	**Chamilo**	**FSKTM Pre-viva Management System**
Type of system	Learning Management System	Learning Management System	Learning Management System	Assessment Management System
Login Module	Yes	Yes	Yes	Yes
Database Connection	Yes	Yes	Yes	Yes
User Authorization	Yes	No	Yes	Yes
Email notification	No	Yes	Yes	Yes
File Submission	Yes	Yes	Yes	Yes

Table 1: (cont.)

User Information	Admin able to add or delete users.	All users can addor update the information.	All users can addor update the information.	Admin able to manage all information of users.
Report Module	No	No	Yes	Yes
Scheduling	No	No	No	Yes

3. Methodology/Framework

This section provide an overview of the merhodology and analysis in this project. The pre-viva management system has for follow the waterfall moall in the system development.

3.1 Waterfall Model

The Waterfall model is a The Pre-viva Management System leveraged by the Waterfall methodology, a structured approach that ensures a systemic and efficient development process through distinct phases. The Waterfall model is one of the SDLC models in developing the flow of the system [2]. Additionally, the waterfall model illustrated the development process into a linear sequential flow which for a process to start the previous process must be finished. This makes the waterfall model easy to follow and understand for the developer [3].

Additionally, the pre-viva management system has provided well-defined requirements and minimal scope that aligned perfectly with the Waterfalls' well-defined stages and deliverables. Table 2 shows the list of each phase with its activities and output which need to be produced during the entire project development.

Table 2: Software Development Activities

Phase	Activity	Output
Requirement Analysis	1. Analysis of the software and hardware requirement 2. Identify the functional and functional requirement	Software and hardware requirements Functional requirement and functional requirement
Design	1. Design the data flow of the system 2. Design the interface of the website 3. Design database	Wireframe Database Design System flow design
Development	1. Develop the website of the Pre-viva management system 2. Develop a database 3. Develop of modules	Database connection Website for administration, student, and supervisor
Testing	1. Conduct system test Conduct user acceptance test	Error and debugging Result of user acceptance test Result of unit test
Maintenance	1. Error solving	Error Fix

3.1 Requirement Analysis Phase

Requirement Analysis is a technical process that can help the developer in determining the need and the condition of the user toward the system. This process is significant as it reduces the chance for developers to develop wrong functions in the system [4]. Additionally, the analysis phase is a phase in which the gathered information will be analysed in detail. With a meeting with the client, the user requirement for the Pre-viva Management System can be identified. Table 3 shows the functional Requirement of the pre-viva managment system

Table 3: Functional Requirement

Module	Function	Users
Login module	• The system should allow user to log into the system with their username and password • The system should alert the user about invalid input with an alert message • The system should redirect the user to their respective dashboard based on their role	• Student • Academic staff • Administrator
User Management Module	• The system should allow the user to create an account • The system should show an error message when the username is duplicate • The system should show an error message when there is an empty field • The system should allow the administrator to edit other user information. • The system should allow the administrator to remove the user from the system • The system should allow users to change their information in the system	• Administrator • Academic Staff • Student
Pre-viva management system	• The system should allow the user to apply pre-viva examination • The system should allow the user to add detail to the pre-viva application • The system should allow the user to schedule the pre-viva examination • The system should allow the user to invite other users to the pre-viva examination	• Student • Supervisor • Administrator
Email Notification	• The system should enable the user to send an email to another user	• Supervisor • Examiner • Chairman
File Submission	• The system should allow users to upload files into the system • The system should allow users to download files from the system	• Supervisor • Examiner • Chairman • Student
Pre-viva Evaluation	• The system should enable users to evaluate the pre-viva examination	• Examiner • Supervisor
Report	• The system should generate an analysis about the results of the student • The system should show the report of the evaluation of the pre-viva examination	• Admin • Supervisor • Examiner • Chairman

Non-functional requirements refer to the performance of the system and the characteristics of the system such as space, time, and dependability[5]. This non-functional requirement was built with 3 components which are performance, operational, and security[6]. Table 4 shows the non-functional requirement of the pre-viva management system.

Table 4: Nun-Functional Requirement

No	Requirement	Description
1	Performance	● The system should be able to be used at any time. ● The function of the system should perform without any error
2	Operational	● The system should be user-friendly. ● The system should be able to work on any web browser. ● The system is only available when there is an internet connection
3	Security	● User can only access the system by using their account. ● All details of the users will be kept in the database

3.1.2 Design phase

The design phase of the waterfall takes the information in the analysis phase to design the pre-viva management system. Moreover, this phase involved defining the system architecture, data structure, and user- friendly interface. Moreover, this phase also designs the data flow of the system such as the context diagram, data flow diagram, and entity relation diagram. The data flow diagram will illustrate the flow of the data in the system while the Entity Relation Diagram will illustrate the flow of data from the system to the database. Additionally, the wireframe interface of the pre-viva management system has been designed through the visualstudio code.

3.1.3 Development Phase

In the development phase, the Pre-viva Management System will be developed according to the design specification by using the visual code as the platform for developing the system. PHP programming has been used in developing the interface and the functionality of the system[7]. Moreover, the database was connected from XAMPP which is a web server. MariaDB is one of the components of XAMPP and is a database management system. So, the pre-viva management system can create a database in XAMPP and use the database to store the information.

3.1.4 Testing Phase

Once the development of the Pre-viva Management System is complete then the testing phase will begin. This phase involves 2 stages which are a system test and a user acceptance test[8]. The system test is conducted to test the Pre-viva Management System. In system tests, the test will be divided into 2 tests which are functional testing and performance testing. Functional testing is for verifying all the system functionalities work as intended while performance testing is for evaluating the system performance under various load conditions. Any error that comes out on the testing will be addressed by the developer to ensure the function of the Pre-viva Management System is usable and does not have any error on each function. Additionally, during this phase, the developer can identify whether the developed system has met the users' needs and requirements[9].

3.1.5 Maintenance Phase

This phase is the phase where the developer will maintain the system to fix the errors that occur that are found during the testing phase and improve the performance of the system[10].

4. Result and Discussion

This section explains the database design, system design, data flow diagram, and interface design of the pre- viva management system. Additionally, this section also describes the system implementation process and the system testing for the Pre-Viva Management System.

4.1 System Design

This section will discuss the system design of the pre-viva management system which is represented in the form of a flow chart. There will be 3 types of users in the pre-viva management system which are admin, supervisor, and student. There will be 2 flow charts showing the flow of the pre-viva examination. The flow chart for applying pre-viva examination is in Appendix A. The flow chart for evaluating the pre-viva examination is in Appendix B

4.2 Data flow Diagram

A data flow diagram is a diagram that explains the flow of the data across the proposed system. Additionally, the data flow diagram also illustrated the flow of the process in terms of input and output. Moreover, it is also

known as a graphical modelling tool that breaks down the system according to the process to help develop the framework of the system [3].

4.2.1 Data flow Diagram level 0

Data flow diagram level 0 shows the whole process with each data flow in the system. Additionally, it also shows the flow of input and output for all entities, system processes, and the database. Appendix E shows the data flow diagram level 0 of the system.

4.2.2 Context Diagram

A context diagram is a diagram that represents the entire system to show the data flow from the entity to the system. In the context diagram, the pre-viva management system consists of 5 entities which are the administrator, student, supervisor, chairman, and examiner. These entities provide different types of data and receive output from the system. The context diagram is shown in Appendix D.

4.3 Database Design

The databse desig is the process of creating and structuring a database of a system.

4.3.1 Entity Relationship Diagram

An entity relationship diagram is a diagram that represents the relationship between each entity. The entities consist of the administrator, supervisor, pre-viva, and student. Each entity shows the database table in the pre-viva management system. Figure 1 shows the entity relationship diagram of the pre-viva management system.

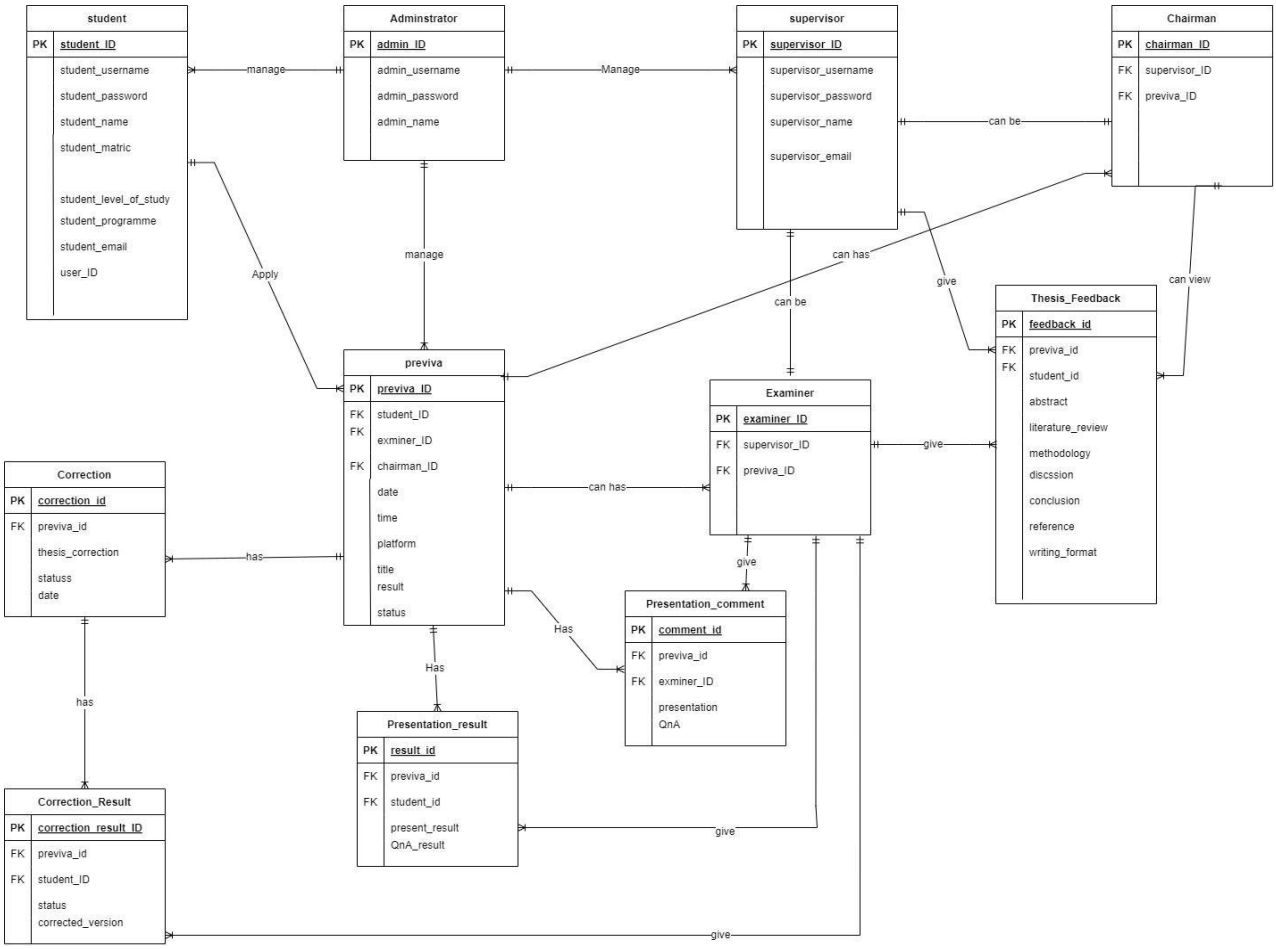


Fig. 1: Entity relationship diagram

4.5 System Implementation

This section shows the interface design and sme security implementation of the system. The software that use in develop the pe-viva management system is visual studio code. This system also had been host by using hosting platform which is 000webhosting. 000webhosting is a free hosting service provider that support database that using phpMyAdmin with MYSQL.

4.5.1 Security Module

The system has implement some security module to enchance the security of pre-viva management system. The security that had implement in the system is hashing, prepare statement, session and prepare statement. Hashing is a process of transforming any given string into another string which no same with the original string. In this project, hashing process is use to prevent the password of the user to be guess by the hacker thus it enchance the security in protect the users information[11]. Next, the term "mysqli_real_escape_string" andprepare statement are a method to prevent the sql injection attack. "mysqli_real_escape_string" will escape all special character in a string before it send the query to MySQL[12]. Next, prepare statemeant also is one of a security method to prevent the SQL injection attack. In the prepare statemen, the sql query was assign parameter with "?" symbol. Then the system will binded the parameter with with the arguments such as "s","d","I","b" and each argument represent different type of parameter. Therefore the input that no fulfill the datatype will no be run [13]. Lastly, the session has been implement when user login into the system. Onve the user logout the system they will no go back to the previous page by clicking back button and the user will always redirect to the login page when they no log into the system. Figure 2 shows the code segment for verify the session and destroy the session. Figure 3 shows the code segment for the prepared statement. Figure 4 shows the code segment for the hashing process.

Fig. 2: Code segment to verify the session (a); Code segment to destroy session (b)

auth.php

```php
<?php
session_start();
include("db_connect.php");
if(!$_SESSION['username']){
header("Location: ../login2.php");
exit(); }
?>
```

(a)

logout.php

```php
<?php
session_start();

// Destroy the session
if(session_destroy()){
// Redirect to the login page or any other page as needed
header('Location: ../PSM/login2.php');
exit();
}
?>
```

(b)

```php
$sql = "SELECT supervisor_email FROM supervisor WHERE supervisor_name = ?";
$stmt = mysqli_prepare($conn, $sql);
mysqli_stmt_bind_param($stmt, "s", $name);
mysqli_stmt_execute($stmt);

$result = mysqli_stmt_get_result($stmt);
$row = mysqli_fetch_array($result);

if ($row) {
    return $row['supervisor_email'];
} else {
    return null;
}
```

```
$previvaID = mysqli_real_escape_string($conn, $_POST['previva_ID']);
$userName = mysqli_real_escape_string($conn, $_POST['name']);
$absComment = mysqli_real_escape_string($conn, $_POST['absComment']);
$introComment = mysqli_real_escape_string($conn, $_POST['introComment']);
$liComment = mysqli_real_escape_string($conn, $_POST['liComment']);
$methoComment = mysqli_real_escape_string($conn, $_POST['methoComment']);
$resComment = mysqli_real_escape_string($conn, $_POST['resComment']);
```

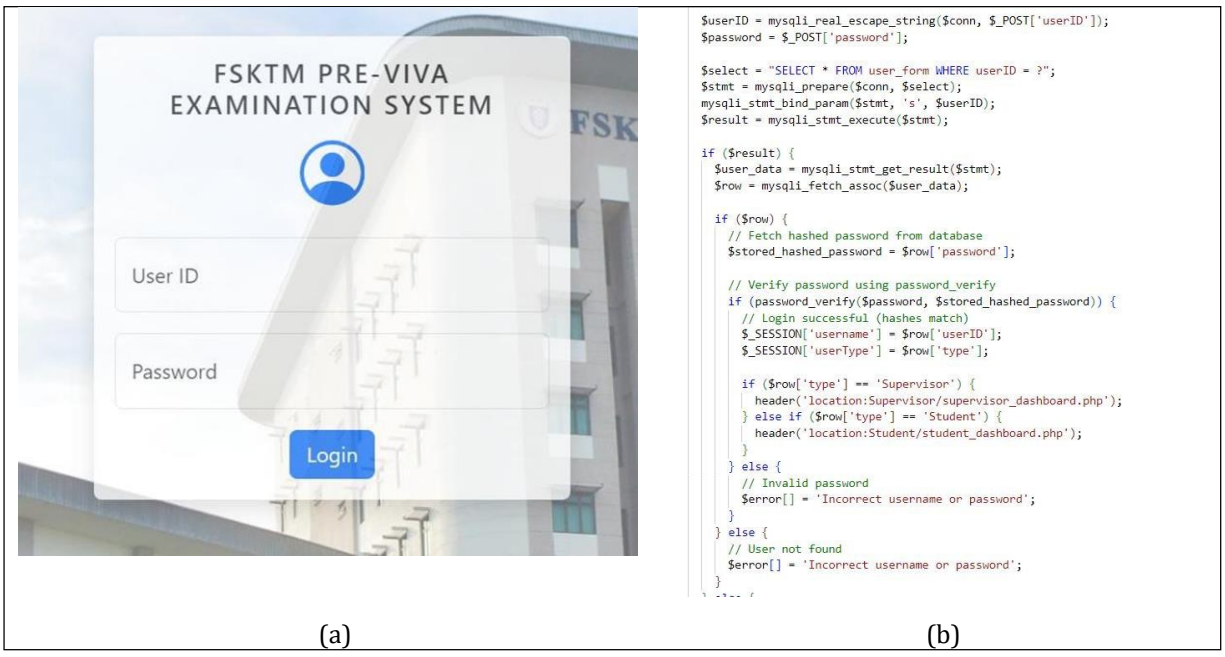
Figure 4: Code of getting the input from the form that submit by the users

```
$hashedPassword = password_hash($newPassword, PASSWORD_DEFAULT);
```

Fig. 4: Code of the hashing to the password

4.5.1 Login Module

This module is about the login process of the user to login into the system. There are 3 type of users which are student, supervisor, and also administrator. Users need to enter their username or userid and also password to enter into the system. Once successful log in, the system will redirect the users to the corresponding dashboard according to the type of user. Figure5(a) shows the login interface for student and supervisor. Figure 5(b) is the code segment for the login module.



(a) (b)

Fig. 5:Login interface for student and supervisor (a); Code segment for login (b)

4.5.2 User Management Module

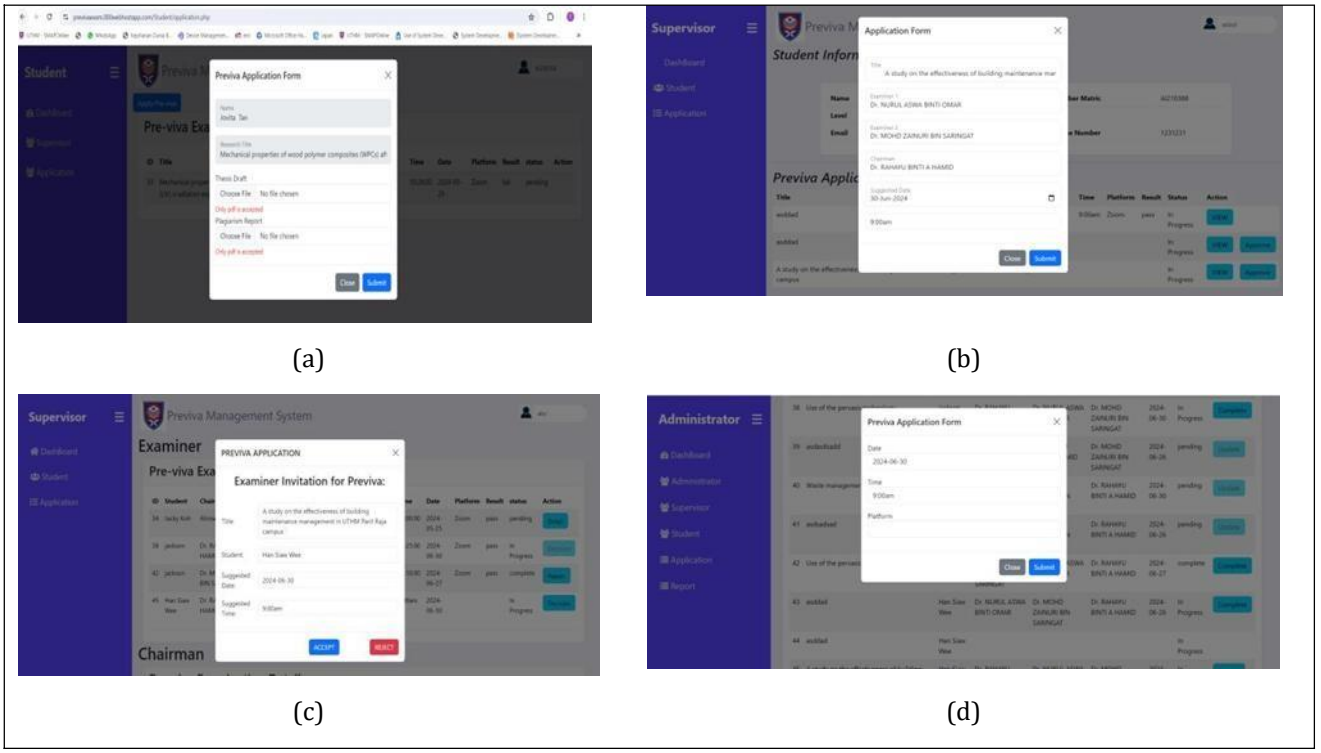
This module describe the precess of creating new user into the system and also remove the user from the system. Super administrator can add new subadmin to help in manage the system. Additionally, super administrator can insert new student and supervisor into the system. Figure 6 (a) shows the code segment for insert new user into the system. Figure 6 (b) shows the interface design for manage the supervisor.

(a) (b)

Fig. 6: Code insert new users (a); interface of manage supervisor (b)

4.5.3 Pre-viva Application Module

This module illustrate the whole process for apply the pre-viva examination. First, student will apply the pre- viva by upload the thesis report and the plagiarism report. Next, supervisor of the student will appoint 2 examiner and chairman to join the pre-viva examination. Supervisor will schedule the date and time for the pre- viva examination. Examaniner and chairman can make decision whether accept or reject the pre-viva examination. When both examiner and chairman accept the invitaion, then admin can set up the platform for the pre-viva examination. Figures 7(a) to 7(d) show the user interfaces for the pre-viva application procedures for the student, supervisor, examiner, chairman, and administrator respectively.



(a) (b)

(c) (d)

Fig.7: Pre-viva Application Page (student);Pre-viva Application Page (supervisor);Pre-viva Application Page (examiner or chairman);Pre-viva Application Page (administrator)

4.5.4 Pre-viva Evaluation Module

The thesis content and the presentation performance of the student will be evaluated by the supervisor and examiner. Examiners and supervisors will evaluate the thesis content and the presentation performance of the student. Additionally, examiners and supervisors also will give comments on each element. Students must get 80 and above marks from the evaluation then they will pass the pre-viva examination. The total marks of the student will be determined automatically when the evaluation process is completed, and the result will be released to all users. Figures 8(a) and 8(b) show the interface for the pre-viva evaluation module.

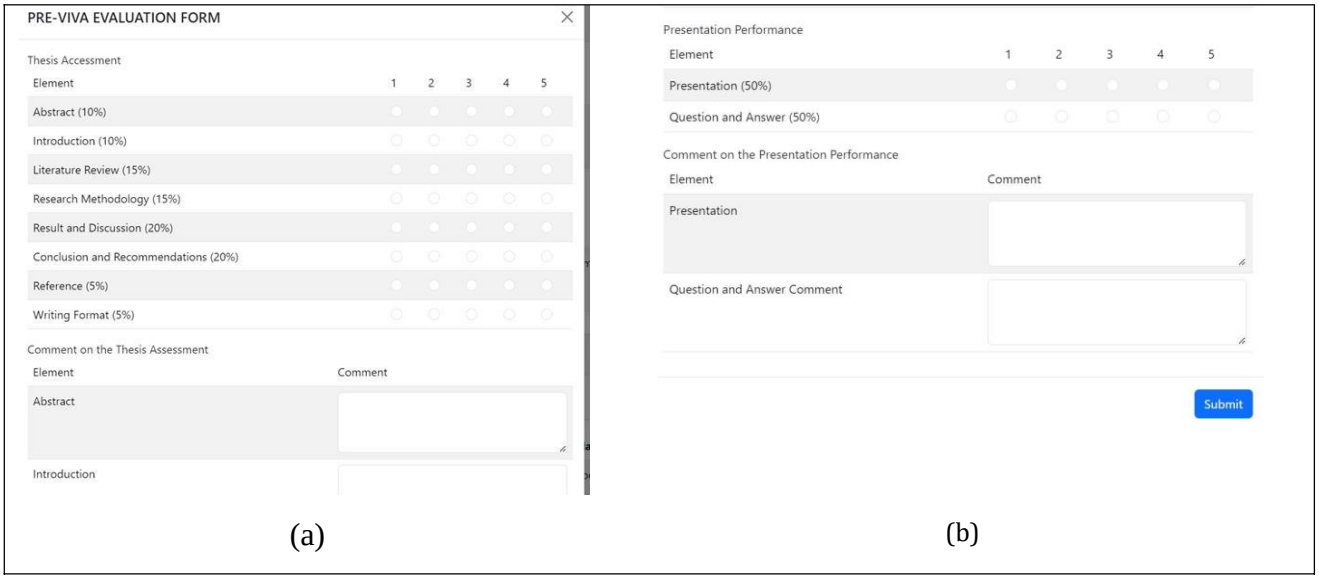


Fig. 8: thesis evaluation interface (a); presentation performance (b)

4.5.5 Email Notification Module

Email notification has been developed in the pre-viva management system. SMTP server has been set up to enable the system to send email via Gmail to the users. Figures 9 to 11 show the interface of the email content sent to the examiner, chairman, and supervisor. Figure 10 is the interface of the invitation email to invite the staff to become examiners or chairman. The invitation email will show the staff the information of the pre-viva examination such as the student's name, the title of the thesis, and the time and date of the pre-viva examination.Figure 11 shows the email content when examiners or chairman accept the previva examination. Figure 9 showsthe rejection email that send to the supervisor with the reject reason from the examiner or chairman. Figure 12 shows the code segment for setting the SMTP server to the pre-viva management system.

Fig. 9: Invitation email content

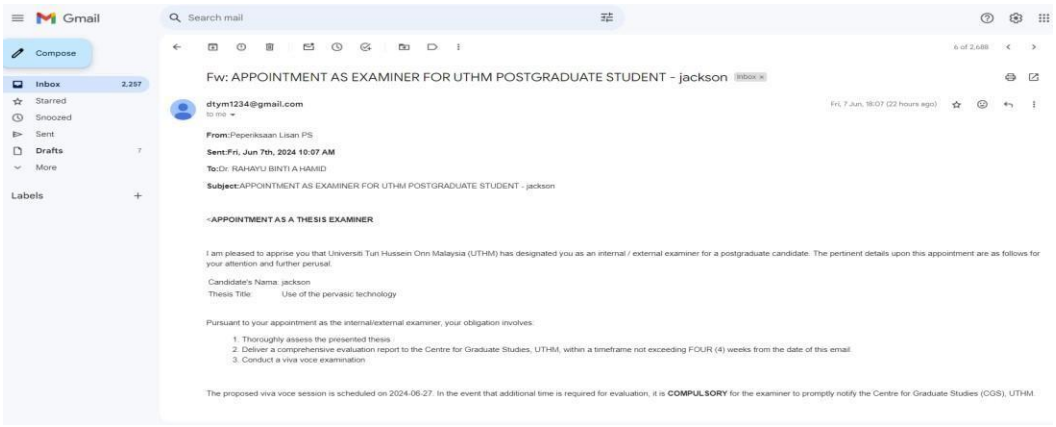


Fig. 10: Appointment Email content

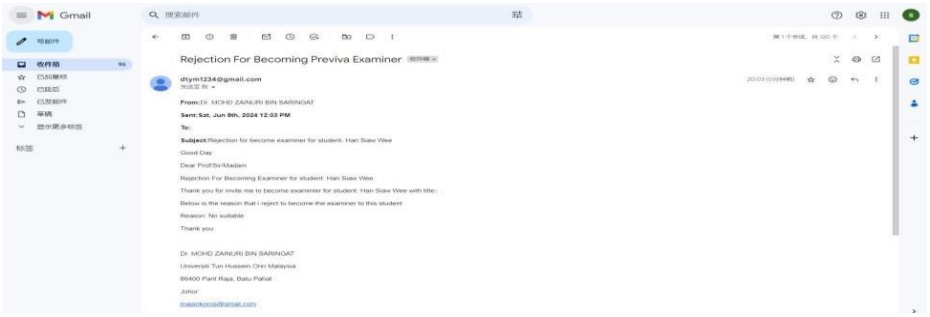


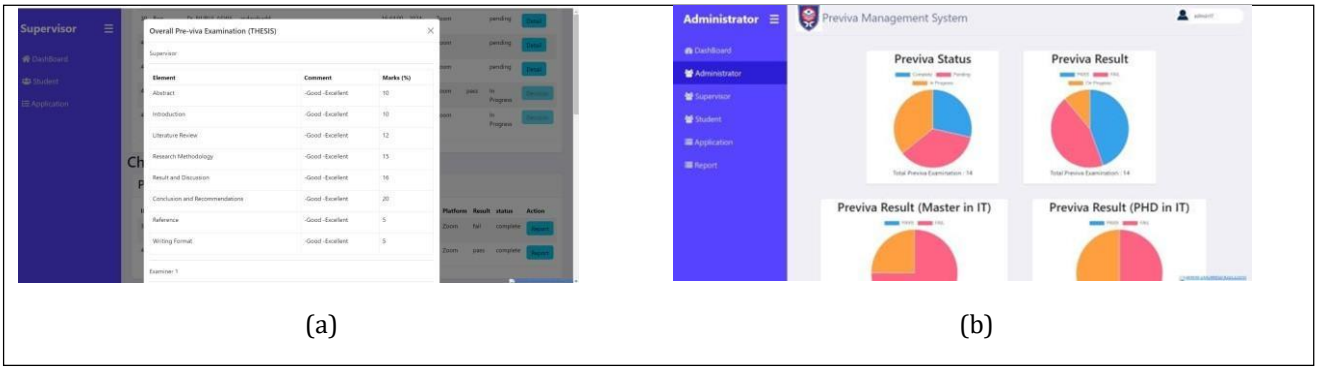
Fig. 11: Reject email content

```php
$mail = new PHPMailer(true);
$currentDate = date('D, M jS, Y h:i A');

$allSent = true;
$mail->SMTPDebug = 0;
$mail->isSMTP();
$mail->Host = 'smtp.gmail.com';
$mail->SMTPAuth = true;
$mail->Username = 'dtym1234@gmail.com';
$mail->Password = 'lxra qagi eobx fkwh';
$mail->SMTPSecure = 'tls';
$mail->Port = 587;

//Send email to chairman
$mail->clearAllRecipients();
$mail->setFrom('dtym1234@gmail.com');
$mail->addAddress($chairmanEmail);
$mail->Subject = "Invitation For Becoming Previva Chairman";
$mail->isHTML(true);
$mail->Body = createEmailContentChairman($chairman_name, $chairman_email, $studentName, $title, $date, $currentDate,$time);
$allSent = $allSent && $mail->send();
```

Fig. 12: Code for seting the SMTP server

4.5.6 Report Module

The report contains the comments and the marks given by the examiner and supervisor during the pre-viva examination. Figure 13 is the report that was only viewed by the administrator which tells the administrator the status of the pre-viva application and the number of fails or passes in the pre-viva examination for studentsin the master's or PhD level. Figure 13(a) and Figure 13(b) show the interface of the evaluation report. Figure 24 is the overall report of the pre-viva examination. It provided an overview of the evaluation to all users.



(a) (b)

Fig. 13: evaluation report interface (a); pie chart report interface (b)

4.6 Testing

The testing step is used to determine if the developed system has fulfilled the requirements of the user. The goal of this testing is to ensure that each module in the system is working properly and does not come out withany mistakes. This test case includes the login module, user management module, and pre-viva application module, . Table 6 shows the testing on the login module

Table 5: Test case of login module

No	Description	Expected	Actual	Result
TEST_100_01	Verify when the username is incorrect or no valid account	The user will receive an error message	An error message is displayed to the user	PASS
TEST_100_02	Verify when the password is incorrect	The user will receive an error message	An error message is displayed to the user	PASS
TEST_100_03	Verify when the usernameis empty	The user will receive an error message	An error message is displayed to the user	PASS
TEST_100_04	Verify when the password is empty	The user will receive an error message	An error message is displayed to the user	PASS
TEST_100_05	Verify when the username and password are correct and match	The user will be redirected to the dashboard of the system according to the role of the users	The user is directed to the dashboard of the system according to the role of the user	PASS

The goal of this test case is to ensure that the insert, update, and delete function in the profile is running properly, and the validation of the input to prevent invalid data or empty to being inserted into the system. Table 6 shows the test case in the profile management module.

Table 6: Test case of user management module

No	Description	Expected	Actual	Result
TEST_200_01	Verify when the administrator does not insert all input fields when adding new users	The administrator will receive an error message	An error message has been displayed to the administrator	PASS
TEST_200_02	Verify whether the user's information has been successfully stored in the database	The administrator will receive a success message	A success message has been displayed to the administrator	PASS

Table 6 ; (cont.)

TEST_200_03	Verify when the users do not insert their information completely to update their profile	The users will receive an error message	An error message has been displayed to the users	PASS
TEST_200_04	Verify whether the users' information has been updated successfully in the database	The users will receive a success message	A success message has been displayed and the data was stored in the database	PASS
TEST_200_05	Verify when administrator want to remove users from the system	A confirmation message will be displayed	A confirmation message has been displayed	PASS

Table 7 shows the test case of the pre-viva application module. The goal of this test case is to ensure that the pre-viva application progress works properly by displaying the message to other users. The upload file function has a specific type of files that are allowed to be submitted in PDF, so there will be an error message when users upload another type of file.

Table 7: Test case of the pre-viva application module

No	Description	Expected	Actual	Result
TEST_300_01	Verify the type of document that uploaded by the student is pdf.	If the upload file is not in PDF, then the student will receive an error message	An error message is displayed to the student when uploading other types of files	PASS
TEST_300_02	Verify whether examiner 1 and examiner 2 have the same value.	If both examiners have the same value, then an error message will be displayed	An error message is displayed to the supervisor	PASS
TEST_300_03	Verify supervisor has selected the examiners, chairman, date, and time, and the data stored in the database	If some input field is empty, then an error message will be displayed. If all input had inserted and the form was submitted, the na success message will be displaye	An error message is displayed when there is an empty input. A success message is displayed when all input have been inserted completly and the form had submitted, a success message was be displayed	PASS
TEST_300_04	Verify the decision of the examiner and chairman on whether to accept or reject the invitation	If the invitation is accepted, then a success message will be displayed. If the invitation is rejected, the reject modal will be shown and ask users to enter the reason	A success message is displayed when the examiners and chairman accept the invitation. The reject modal was shown when the invitation was rejected.	PASS
TEST_300_05	Verify the application status once the administrator sets the platform of the pre-viva examination	When the administrator has set up the platform, the status of the pre-viva application will change to "pending"	The status of the pre-viva examination has changed to "pending" once the platform has been set up	PASS

5. Conclusion

In conclusion, the pre-viva management system had been fully develop in the end of the project. The web-base pre-viva management system become a helpful tool that help the supervisor,, and administrator to manage the the pre-viva application. Additionally, the web-based pre-viva management system had imprve the efficiency of the pre-viva examination. It enchance the process for applying and updating the pre-viva application. Although there are some limitation of the pre-viva management system, but with the future development of the pre-viva management system, the limitation of the system will be overcome.

Acknowledgement

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

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Derrick Tee Yee Meng, Hairulnizam Mahdin;* ***data collection:*** *Derrick Tee Yee Meng;* ***analysis and interpretation of results:****Derrick Tee Yee Meng, Hairulnizam Mahdin;* ***draft manuscript preparation:*** *Derrick Tee Yee Meng, Hairulnizam Mahdin. All authors reviewed the results and approved the final version of the manuscript.*

An author name can appear multiple times, and each author name must appear at least once. For single authors, use the following wording:

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

References

[1] Zoe Stephenson, Amy Jackson & Victoria Wikes (2023) Student experiences of the 'closeddoor' PhD and doctorate level viva voce: a systematic review of the literature, Assessment & Evaluation in Higher Education, DOI: 10.1080/02602938.2023.2282941

[2] Senarath, U. S. (2021). Waterfall methodology, prototyping and agile development. *ResearchGate*. https://doi.org/10.13140/RG.2.2.17918.72001

[3] Astuti, L., & Supriatno, S. (2021b). System requirement analysis for the initial development of the information system of review and assessment process in P2STPFRZR- BAPETEN. *Nucleation and Atmospheric Aerosols*. https://doi.org/10.1063/5.0058975

[4] A. Ratnawati, E. Kartikasari, B. Rustandi, B. Susanto, A. Setiawan and K. A. Munastha, "A Web-Based Accounting Information System Application using CodeIgniter Framework: (A Case Study Approach)," 2022 16th International Conference on Telecommunication Systems, Services, and Applications (TSSA), Lombok, Indonesia, 2022, pp. 1-6, Doi: 10.1109/TSSA56819.2022.10063881.

[5] Chong, H., & Διαμαντόπουλος, A. (2020). Integrating advanced technologies to uphold security of payment: Data flow diagram. *Automation in Construction*, *114*, 103158. https://doi.org/10.1016/j.autcon.2020.103158

[6] G. A. Alencar, F. V. De S. Oliveira, J. Da Silva Correia-Neto and M. M. Teixeira, "NonFunctional Requirements In Health Information Systems:" 2019 14th Iberian Conference on Information Systems and Technologies (CISTI), Coimbra, Portugal, 2019, pp. 1-5, doi: 10.23919/CISTI.2019.8760720.

[7] Gupta, S., & Singh, A. (2022). Secure thesis management using PHP in the Pre-viva Management System. International Journal of Information Technology, 14(4), 18-27.

[8] Jones, B. (2023). Enhancing the pre-viva process with PHP: A case study of the Pre-viva Management System. International Journal of Education and Technology, 20(3), 1-12.

[9] J. Mantik, Apriana, V., & Fauziah, S. (2021). Applying Waterfall Method on Sales Information System. *Journal Mantik*, *5*(2), 820–826. https://doi.org/10.35335/mantik.vol5.2021.1380.pp820-826

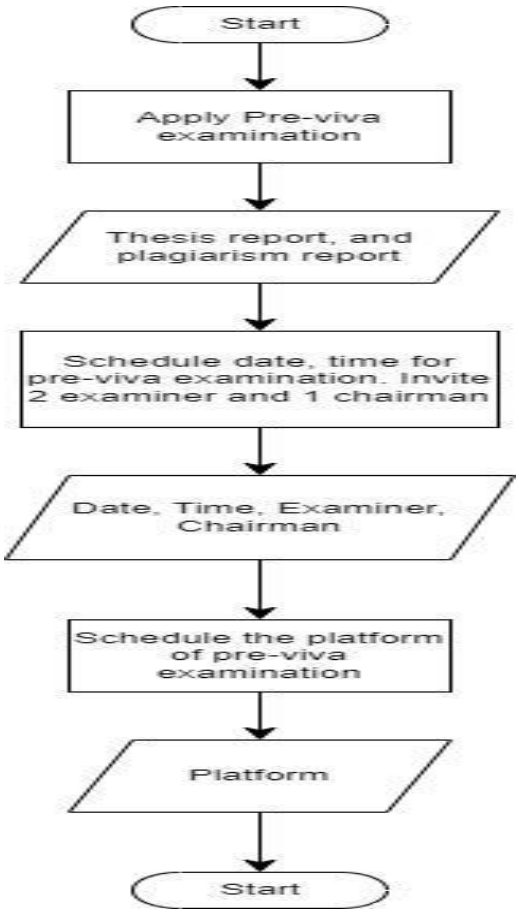
[10] K. Shehadeh, N. Arman and F. Khamayseh, "Semi-Automated Classification of Arabic U ser Requirements into Functional and Non-Functional Requirements using NLP Tools," 2021 International Conference on Information Technology (ICIT), Amman, Jordan, 2021, pp. 527532, doi: 10.1109/ICIT52682.20 21.9491698.

[11] A. Zola, "What is hashing and how does it work?," *TechTarget*, Jun. 2021. https://www.techtarget.com/searchdatamanagement/definition/hashing

[12] "mysql_real_escape_string SQL injection - Correct Usage and Attacks." https://www.sqlinjection.net/advanced/php/mysql-real-escape-string/

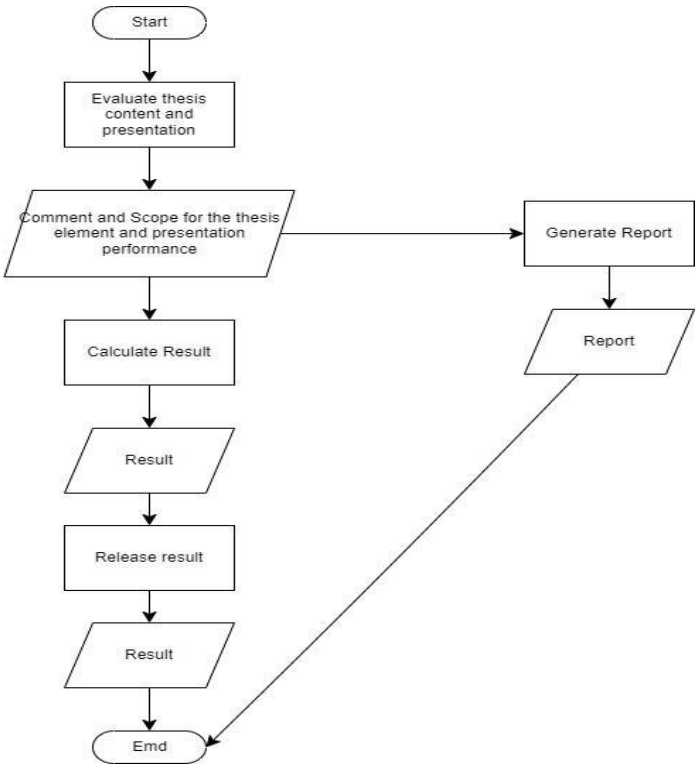
[13] W3Schools, "PHP MySQL Prepared Statements," *www.w3schools.com*, 2023. https://www.w3schools.com/php/php_mysql_prepared_statements.asp

Appendix A: Flow chart of apply Pre-viva Application

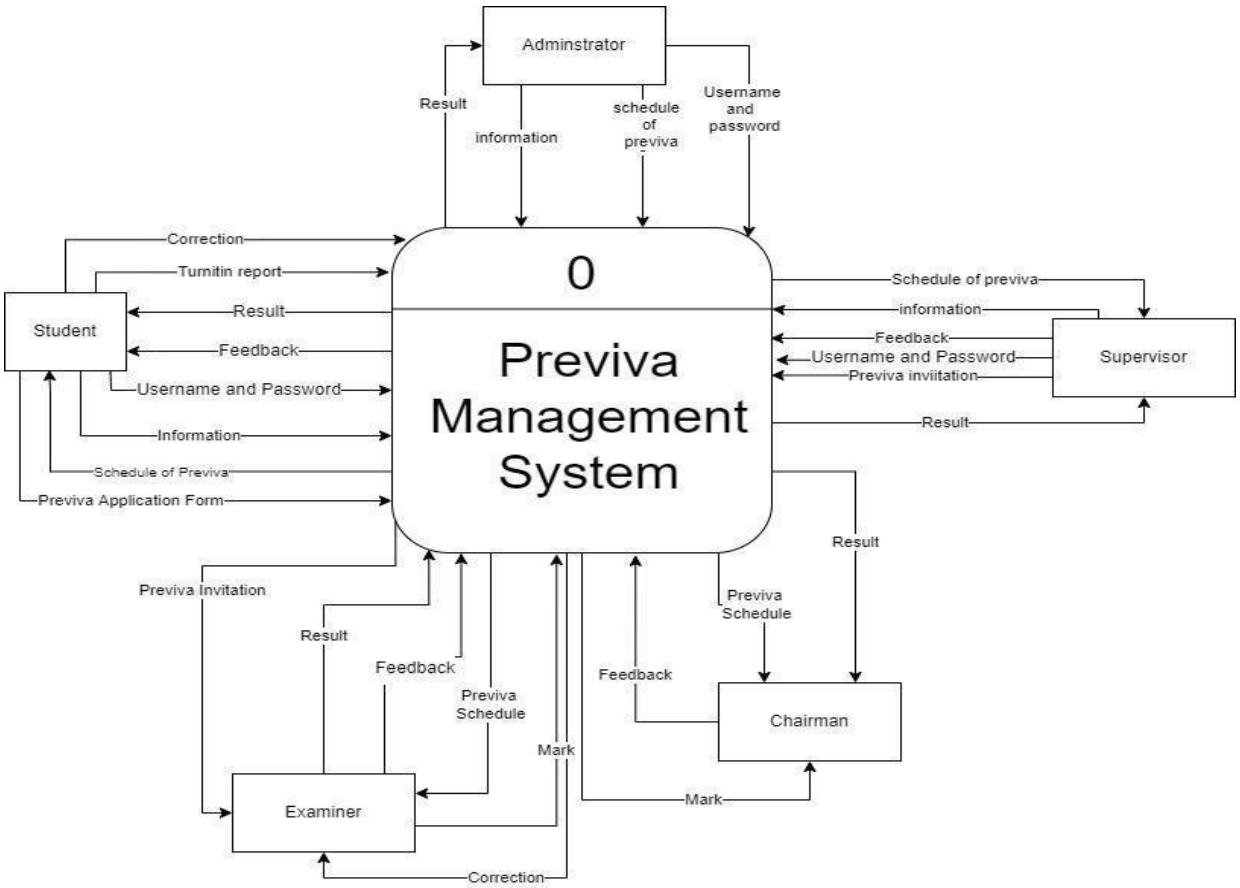


Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 273-290

Appendix B: Flow chart of evaluate pre-viva examination



Appendix D: Context Diagram



Appendix E: Data Flow Diagram Level 0