**Chapter 4**

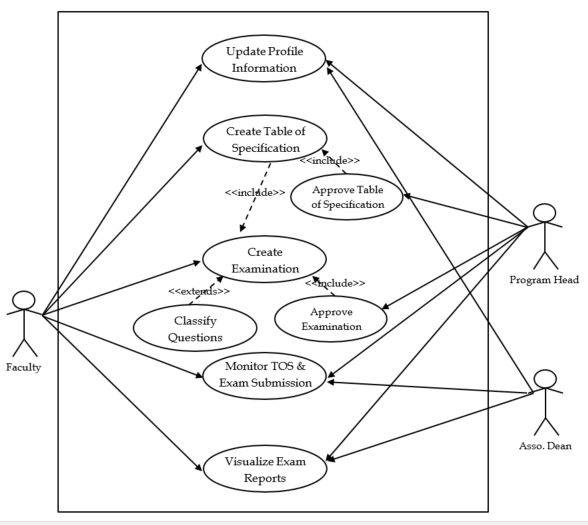
**RESULTS AND DISCUSSION**

**Project Technical Description**

The Examination Authoring System, as the product of the Application of Machine Learning and Natural Language Processing in Classifying Examination Questions according to Revised Bloom’s Taxonomy, presents an interactive system that can accommodate faculty members of the MSC – Institute of Information Systems and Technology, in creating Table of Specification and Major Examination in a centralized web-based platform. The following are the functional specifications of the developed system based on the projects’ objectives:

1. *Web-based Authoring System*. The Web-based system is capable of authoring two important documents, which are the Table of Specification and the Major Examination. It can help create the said documents by having a standardized web interface. As an authoring tool, the system follows the prescribed design and layout of the output document. Faculty members who are the primary user of the system can experience the automated process of creating their TOS and examination. The system provides faculty members to conveniently add topics that are included in the TOS. It can automatically compute the number of items per particular topic based on the number of hours entered in the TOS. As for the type of test that will be included in the examination, faculty members can add a type of tests that is applicable to the cognitive skills of Bloom’s Taxonomy. Automatic computation of the number of items per type of test and its percentage equivalent can also be found in the system. Once the created TOS is approved, faculty members may start creating their examinations. In creating an examination, only the type of tests present in the created TOS shall be added to the exam. Several types of tests are already classified according to the proper cognitive level. Questions under the type of Multiple Choice and Essay are the ones that would be automatically classified to what cognitive level they belong. The system provides a facility to edit or reconstruct the created questions. Proper formatting of the type of tests is also provided in the system. The system also provides a facility to print the created TOS and examination.
2. *Management of TOS and Examination*. The created TOS and examination of faculty members can be submitted to their respective Program Heads. Program Heads, as secondary users, can access the system for the monitoring of TOS and examination submissions. The system provides a facility to approve and disapprove for revision of the submitted TOS and examination. A list indicating the names of the faculty members who have submitted their TOS and examination is provided in the system. Different status for creating TOS and examination is also provided, such as Draft, Submitted, and Approved. This functionality can help program heads determine faculty members who are not yet creating and/or submitting their TOS and examination.
3. *Question Classification*. The question classification is the main feature of the system. It was made possible by utilizing machine learning techniques and natural language processing. This function is employed using Python programming language. The script that holds its program is called inside the web-based system created with the Laravel framework. The python script is powered by a naïve bayes classifier caller MultinomialNB(). It is a powerful algorithm that is for text data analysis and classification. Since it implements the Bayes Theorem, it uses probability to determine a possible prediction or classification of newly created questions. This classifier was trained using a question dataset that underwent text pre-processing using NLP. The training set of the question dataset has an accuracy of 97%. When tested, it provides a correct classification of the questions based on the cognitive level of Bloom’s Taxonomy. Question Classifier is visible on the system when faculty members create questions for their exams. It will automatically provide a status of their classification beside the created questions.
4. *Exam Question Validation*. This validation function refers only to the process of checking the creation of an exam based on what is provided in the created TOS. The system shall verify whether a particular question with its classification is parallel to the item numbers in the TOS. The TOS shall serve as the guidelines in providing the questions as entries in the examination.
5. *Reports Visualization*. Examination reports and statistics are presented in the system. It consists of a graphical representation depicting the percentage distribution of the number of questions classified according to the levels of Bloom’s Taxonomy. Exam reports also include the categorization of the examination questions for the Higher Order Thinking Skills and Lower Order Thinking Skills.

**Use Case of the System**



**Figure 4.1** *Use Case Diagram of the Developed System*

**User Story**

1. **Use Case Title**: Update Personal Information

**Description**: The Faculty Member/Program Heads/Assoc. Dean can update his/her personal information in the system.

**Primary Actor**: Faculty Member/Program/Assoc. Dean

**Goal**: To allow the users to update his/her personal information when needed.

**Pre-Conditions**: Before doing an update, the user must log in to the system.

**Post-Conditions**: The users’ personal information is updated.

**Table 4.1** *Use Case Scenario of Updating the Personal Information*

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. Access the system.  3. Update the personal  information. | 2. Validates user account.  4. Updates the personal  information.  5. Display confirmation message. |

It is shown in table 4.1 the processes how to update the personal information of the Faculty Members, Program Head, and Associate Dean. This process is relevant, especially in case there is a change in the administration of the system. The main role of the system administrator is to manage the system’s settings and maintenance.

1. **Use Case Title**: Create Table of Specification

**Description**: The Faculty can create a table of specification for a particular subject.

**Primary Actor**: Faculty

**Goal**: To create a table of specification

**Pre-Conditions**: The Faculty should have an assigned subject before he/she can create a TOS.

**Post-Conditions**: A table of specifications is created and submitted to the program head.

**Table 4.2** *Use Case Scenario of Creating the Table of Specification*

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. Access the system  3. Select Table of Specification  5. Select Add Specification  7. Enter Specification Details  and Click Save button.  9. Enter Topics and Number of  Hours and Click Save button.  11. Select Cognitive Skill to be  added in the TOS.  13. Enter desired Item Numbers  and Click Save button.  15. Enter all the type of test and  item numbers needed to complete the TOS.  17. Clicks the Submit Button | 2. Validates user account.  4. Display List of TOS Webpage  6. Display Add Specification  Webpage  8. Display confirmation message.  10. Display confirmation and  automatically computes number of items per added topic.  12. Display Type of Test  depending on the selected cognitive skill.  14. Display confirmation message  and the entered type of test and item numbers. Automatically computes the No. of Items and its equivalent percentage.  16. Display Submit button.  18. Display confirmation message  and submits the TOS to the program head. |

1. **Use Case Title:** Check and Approve Table of Specification

**Description**: The Program Head shall check and approve the submitted TOS of Faculty members.

**Primary Actor**: Program Head

**Goal**: To check and approve the submitted TOS of Faculty members.

**Pre-Conditions**: The faculty needs to create and submit TOS first.

**Post-Conditions**: Approved the submitted Table of Specification.

**Table 4.3** *Use Case Scenario of Checking and Approval of TOS*

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. Access the system  3. Select Table of Specification  5. Select and View Submitted  TOS.  7. Approve created TOS. | 2. Validates user account.  4. Display List of Submitted  Table of Specification.  6. Display selected TOS.  8. Display confirmation message. |

1. **Use Case Title**: Create Examination

**Description**: The Faculty Shall Create the Examination based on the TOS he/she created.

**Primary Actor**: Faculty

**Goal**: To create examination and classify each question according to Bloom’s Taxonomy.

**Pre-Conditions**: The program head should have approved the Table of Specification.

**Post-Conditions**: Created and submitted examination and classified each question.

**Table 4.4** *Use Case Scenario of Creating Examination*

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. Access the system  3. Select Examination  5. Enter Examination Details  and Clicks the Save Button.  7. Click Add Questions Button  9. Select type of test displayed  in the drop down menu.  10. Enter item number and type  the question.  12. If the type of test is Multiple  Choice, enter desired  choices for each number.  13. Click Save Button.  16. Completes all the question  per type of test.  17. Clicks the submit button. | 2. Validates user account.  4. Display Create Examination  Interface.  6. Display Interface in creating  examination. Generates the  exam parts based on what  were provided in selected  TOS.  8. Display pop-up interface for  adding questions.  11. Display Grammarly icon, if  there are grammatical errors  on the entered question.  14. Display confirmation message  and added question per type  of test.  15. Automatically analyze and  display the classification of  the added question based on  the proper Cognitive Level the  belong.  18. Display confirmation message  that the examination has been  submitted. |

1. **Use Case Title:** Approve Examination

**Description**: The Program Head shall check and approve the submitted examinations.

**Primary Actor**: Program Head

**Goal**: To approve the submitted examination.

**Pre-Conditions**: Created examination must be submitted.

**Post-Conditions**: The examination is approved and ready for printing.

**Table 4.5** *Use Case Scenario on the Approval of Examination*

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. Access the system  3. Click on the Examination  5. Select the examination that  needs to be approved. | 2. Validates user account.  4. Display the list of submitted  examination.  6. Display the created |
| 7. Read through the  examination. Check if there  are parts that needs to be  revised.  8. Click on For Revision  Button, if there should be  some changes on the  examination.  10.Click Approve button, if the  exam has no revisions to be  made. | examination.  9. Display pop-up window for  Revision of examination.  11.Display approval confirmation  message. |

1. **Use Case Title**: Monitor Table of Specification & Exam Submission

**Description**: The Program Head and Associate can monitor the submission of the TOS and Examination.

**Primary Actor**: Program Head, Associate Dean

**Goal**: To monitor the submission of TOS and examination.

**Pre-Conditions**: TOS and Examination must be created, submitted, and approved in the system.

**Post-Conditions**: Monitored submission of TOS and examination.

**Table 4.6** *Use Case Scenario of Monitoring TOS & Exam Submission*

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. Access the system  3. For TOS: Click on Table of  Specification  5. For Exam: Click on the  Examination | 2. Validates user account.  4. Display list of drafted,  submitted and approved TOS.  6. Display list of drafted,  submitted and approved  Examination. |

1. **Use Case Title**: View Exam Reports

**Description**: The system users may view examination report.

**Primary Actor**: Faculty, Program Head, Associate Dean

**Goal**: To view examination report.

**Pre-Conditions**: The examination should be completed first.

**Post-Conditions**: Displayed the exam reports.

**Table 4.7** *Use Case Scenario of Viewing Exam Reports*

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. Access the system 2. Click on Examination 3. Select the desired Examination. 4. Click on Statistics | 1. Validates user account. 2. Display list of completed examination. 3. Display created examination. 4. Display graphical report on the percentage distribution of questions per cognitive levels of Blooms Taxonomy. |



**Figure 4.2** *Use Case of the System Administrator*

1. **Use Case Title**: Add Academic Year

**Description**: The System Administrator can add academic year to the system.

**Primary Actor**: System Administrator

**Goal**: To add and update an academic year.

**Pre-Conditions**: None

**Post-Conditions**: Academic year is added and/or updated.

**Table 4.8** *Use Case Scenario of Adding an Academic Year*

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. Access the system  3. Add Academic Year | 2. Validates user account.  4. Display confirmation  message. |

The above table shows how academic year can be added/updated in the developed system. The System Administrator needed to access the system before performing the process.

1. **Use Case Title**: Add Program

**Description**: The System Administrator can add program to the system.

**Primary Actor**: System Administrator

**Goal**: To add and update a program.

**Pre-Conditions**: None

**Post-Conditions**: Program is added and/or updated.

**Table 4.9** *Use Case Scenario of Adding a Program*

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. Access the system  3. Select Academic Year  5. Add Program | 2. Validates user account.  4. Display confirmation message. |

1. **Use Case Title**: Add Section

**Description**: The System Administrator can add section to the system.

**Primary Actor**: System Administrator

**Goal**: To add and update a section.

**Pre-Conditions**: None

**Post-Conditions**: New section is added to the system.

**Table 4.10** *Use Case Scenario of Adding a Section*

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. Access the system   3. Add Section. | 1. Validates user account. 2. Display confirmation   message. |

1. **Use Case Title**: Add Subject

**Description**: The System Administrator can add a subject to the system.

**Primary Actor**: System Administrator

**Goal**: To add a new subject to the system.

**Pre-Conditions**: A program should be added first in the system.

**Post-Conditions**: New subject is added to the system.

***Table 4.11*** *Use Case Scenario of Adding a Subject*

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. Access the system   3. Select Program  4. Add Subject | 1. Validates user account.   5. Display confirmation  message. |

1. **Use Case Title**: Register User Account

**Description**: The System Administrator registers a new user account to the system

**Primary Actor**: System Administrator

**Goal**: To register a new account user to the system.

**Pre-Conditions**: None

**Post-Conditions**: Registered new user account to the system.

**Table 4.12** *Use Case Scenario of Registering a New User Account*

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. Access the system 2. Register New User   Account. | 1. Validates user account. 2. Display confirmation message. |

1. **Use Case Title**: Assign Subject

**Description**: The System Administrator may assign a subject to a particular section.

**Primary Actor**: System Administrator

**Goal**: To assign a subject to a particular section.

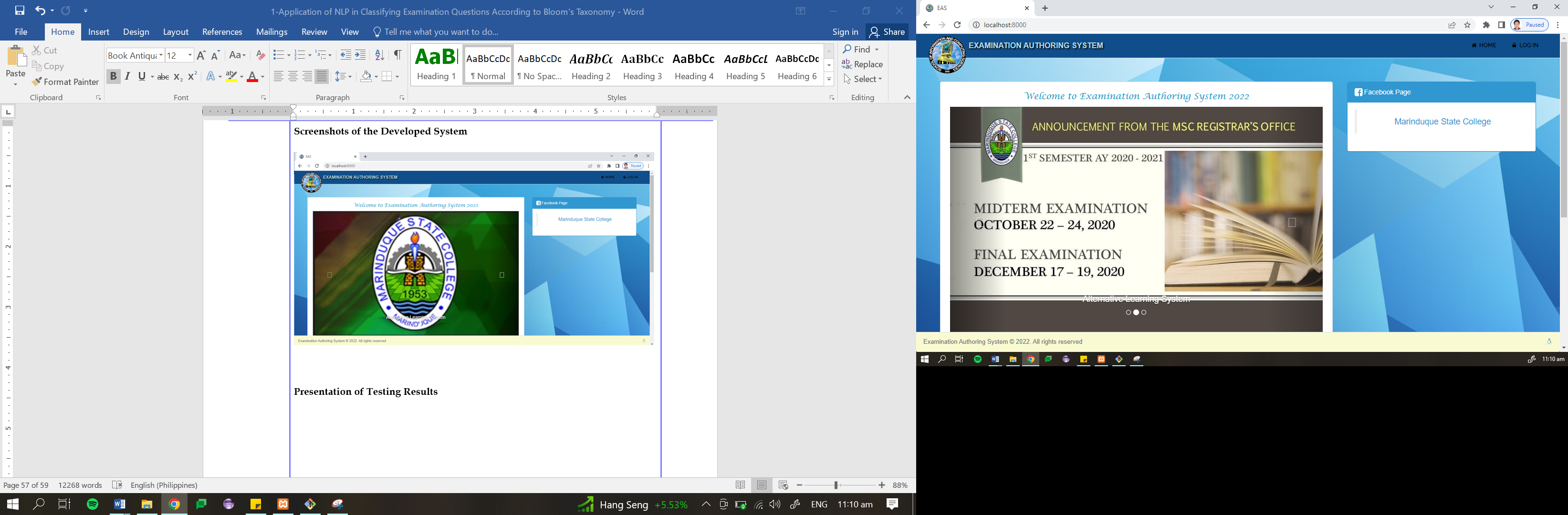
**Pre-Conditions**: A section should be added first in the system.

**Post-Conditions**: A subject was assigned to a particular section.

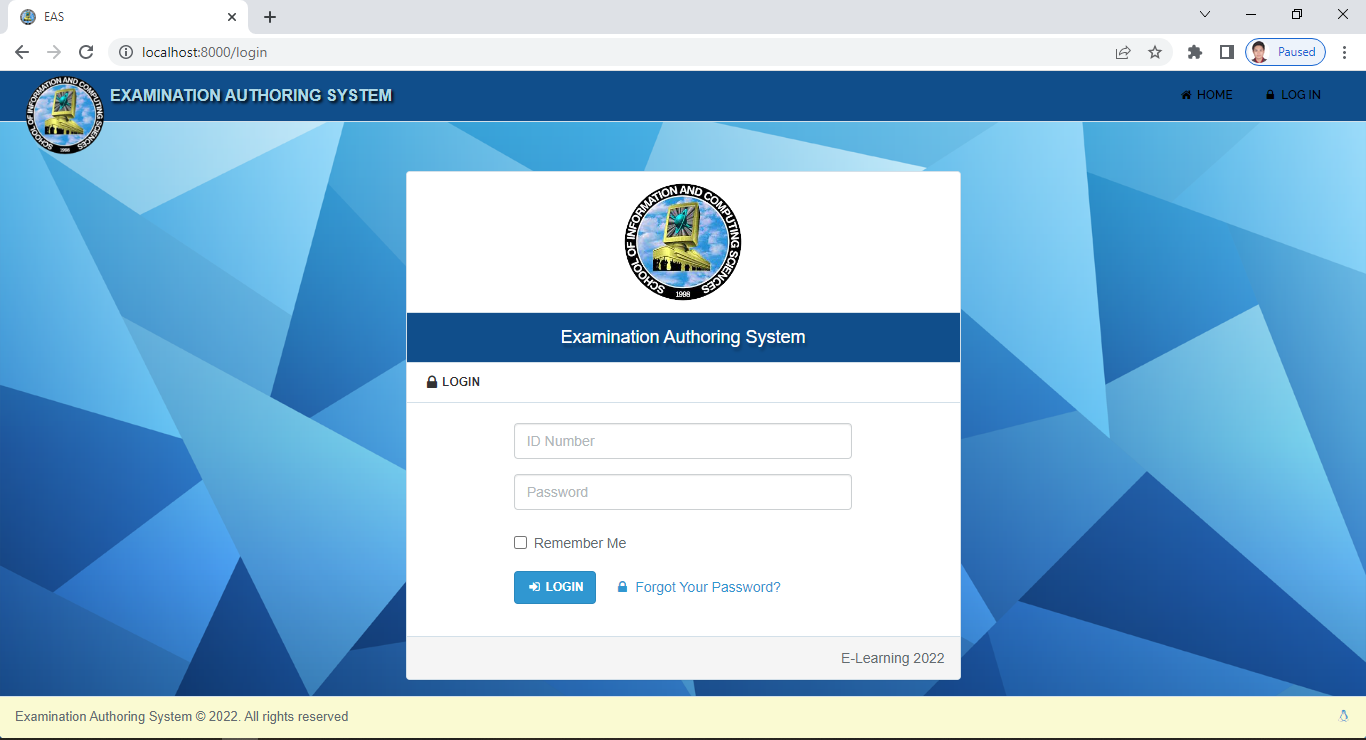
**Table 4.13** *Use Case Scenario of Assigning a Subject to Faculty*

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. Access the system   3. Select Section  4. Assign Subject | 1. Validates user account.   5. Display confirmation  message. |

**Screenshots of the Developed System**

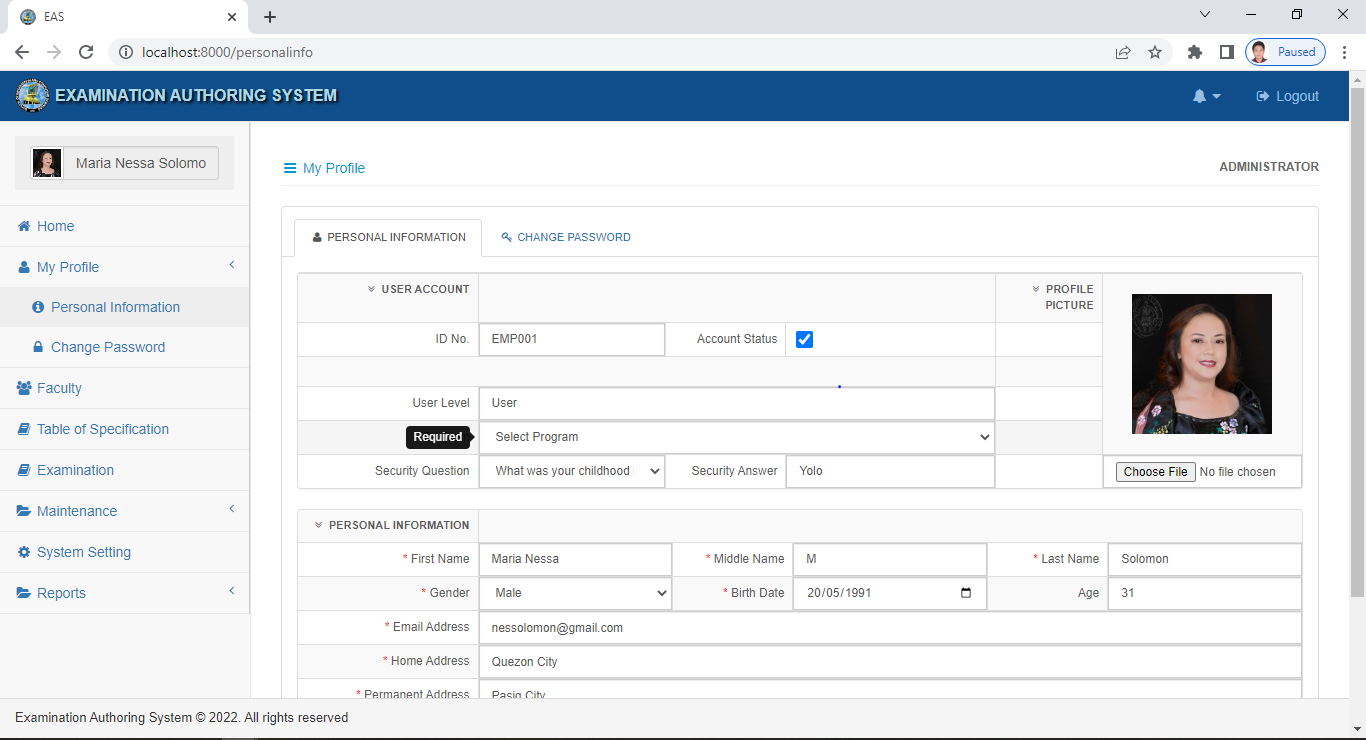


**Figure 4.3** *Home Page of the Examination Authoring System*



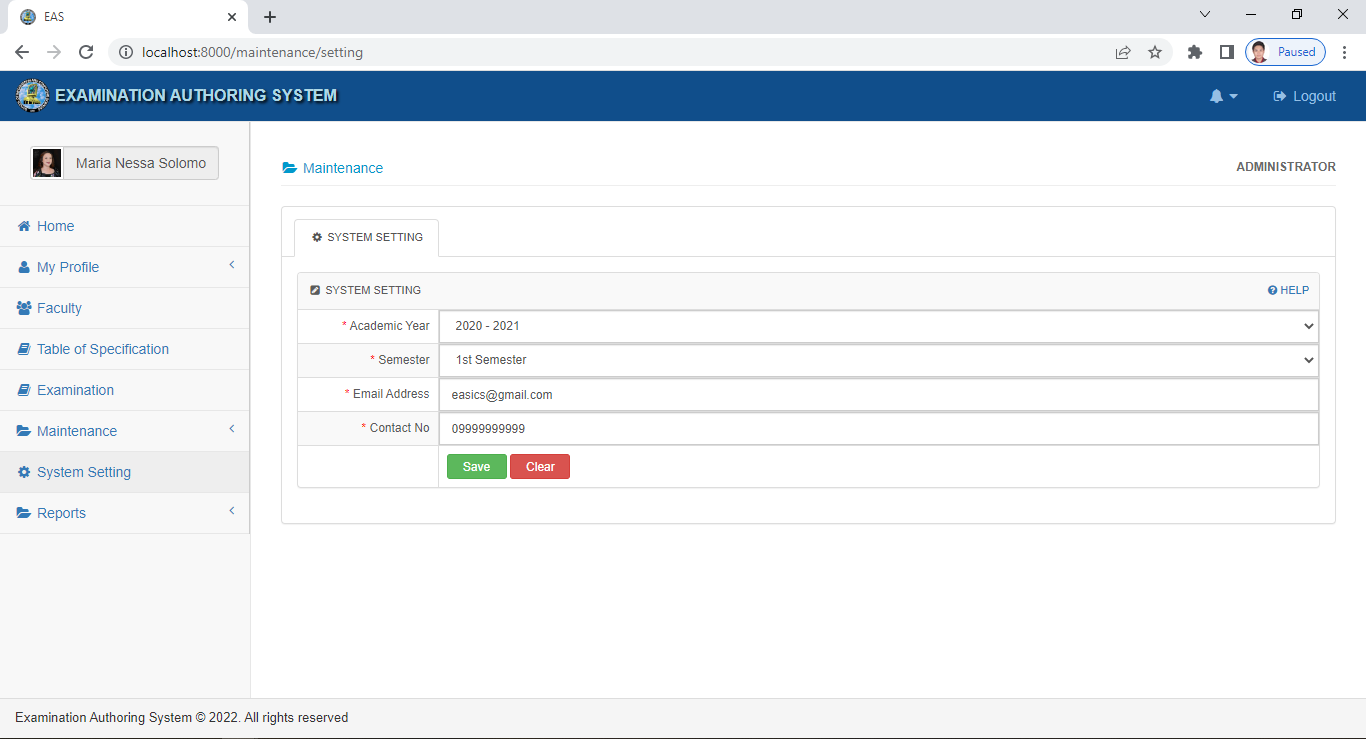
**Figure 4.4** *Log-in Page of the Examination Authoring System*

The developed system has three types of users, the System Administrator or the Dean, the Program Head, and the Faculty. Figure 4.1 serve as the Landing page of the system, while Figure 4.2 is used to access the system. User accounts are created solely by the System Administrator, and the owner of the user may change their password if they want to.



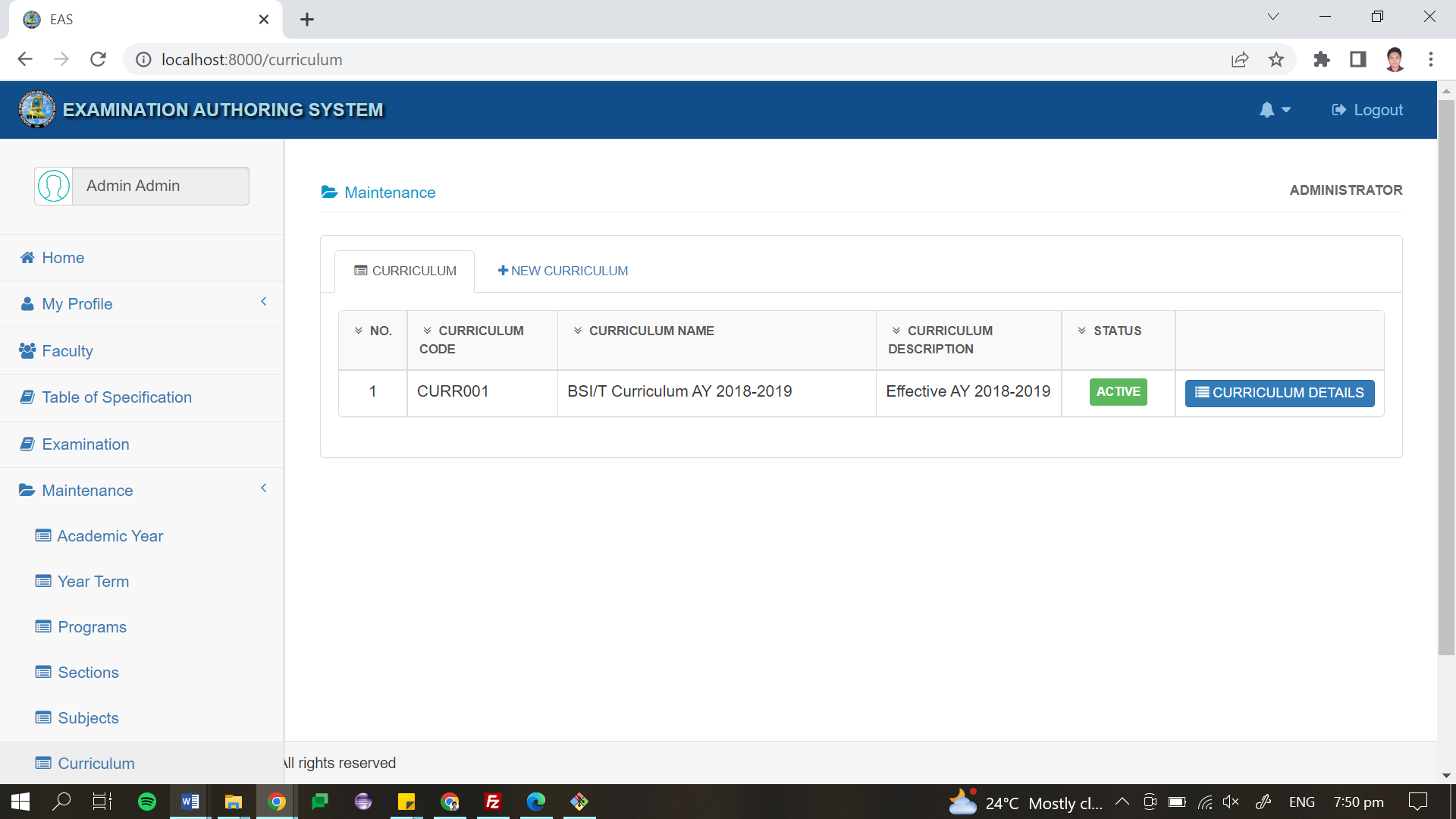
**Figure 4.5** *User Interface of the System Administrator (Dean)*

As shown in Figure 4.5, aside from creating a user account, the System Administrator is also capable of setting up the academic year as well as the active semester and contact details of the Institute (Figure 4.6), manage curriculum (Figure 4.7) to the system, add section (Figure 4.8), add program offerings and assign faculty to the subject (Figure 4.10).



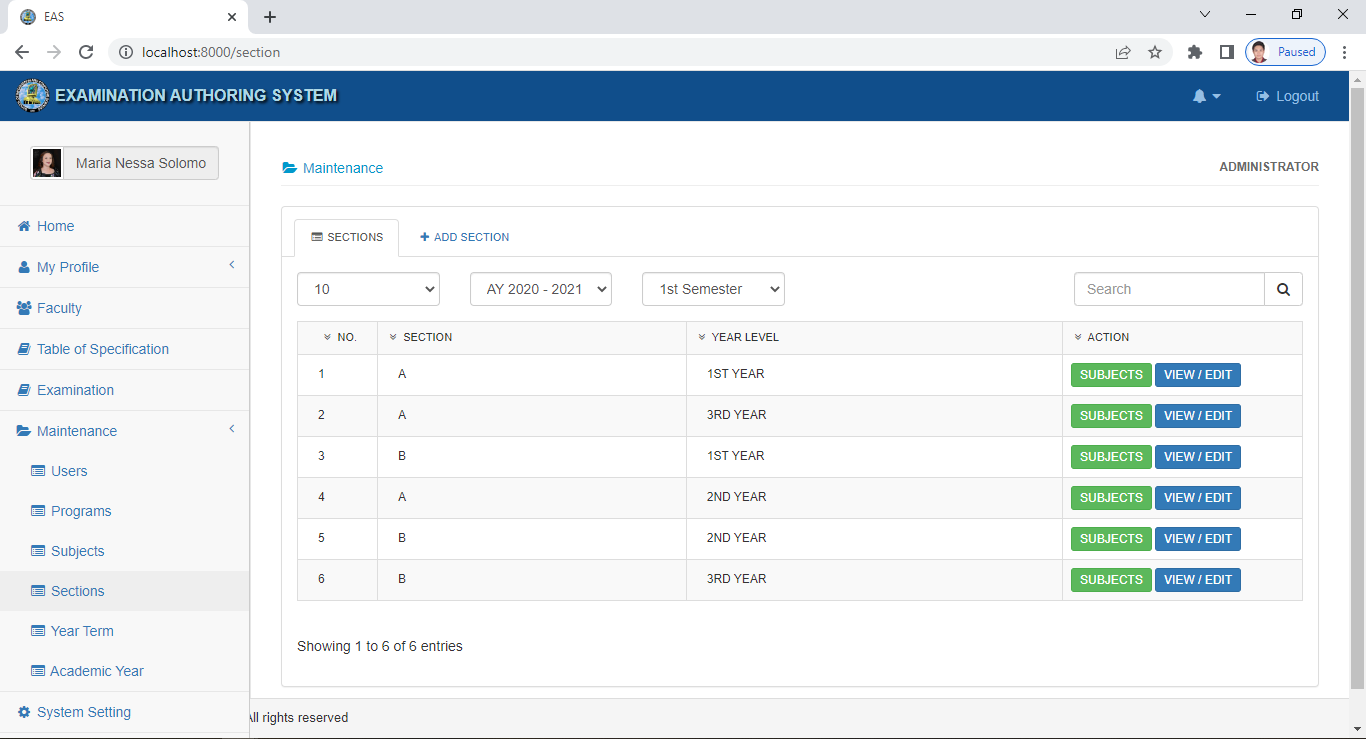
**Figure 4.6** *System Settings*

As shown in Figure 4.6, the administrator can set and change the system settings by setting the active Academic year and Semester that would be used by the system throughout its transactions. Email Address and Contact Number for the examination authoring system could also be set in this page.



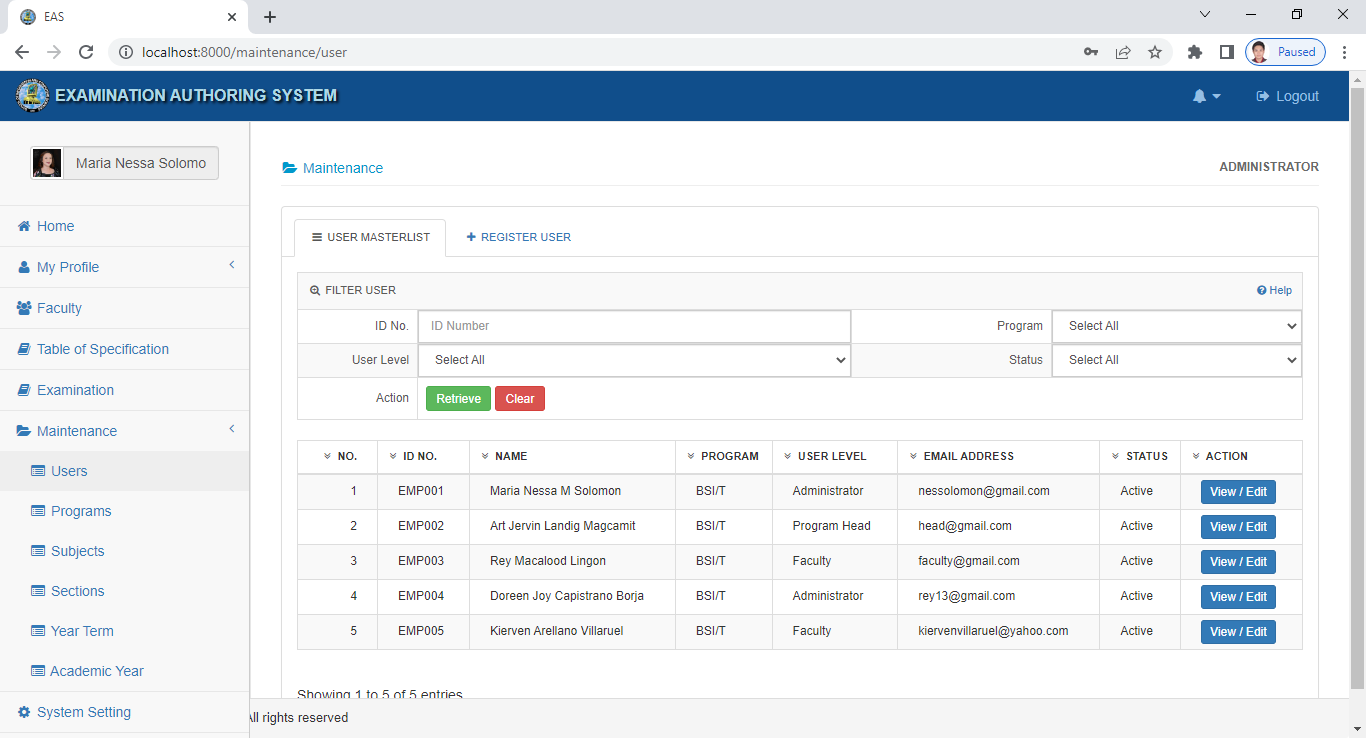
**Figure 4.7** *Curriculum Maintenance*

Figure 4.7 shows the Curriculum Maintenance page. Saved curriculum information can be found on this webpage. It also provides a tab that allows administrator to add a new curriculum if there is any. Different subject information per year and semester can be viewed when the user clicks on the Curriculum Details button provided for each curriculum.



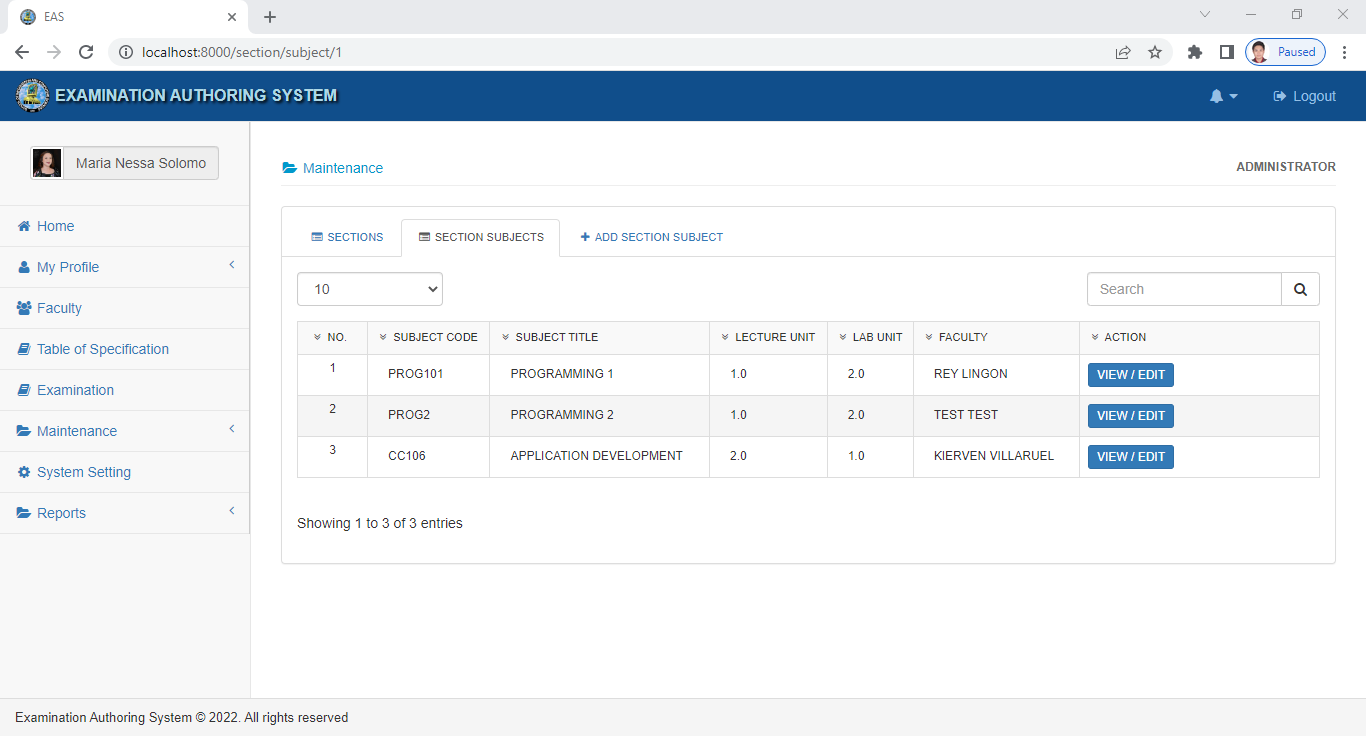
**Figure 4.8** *Setting Up of Class Sections*

Figure 4.8 shows the webpage to where the administrator can add sections per year level. This page also provides the means for administrator to view the different subjects in each section by clicking on the Subjects button.



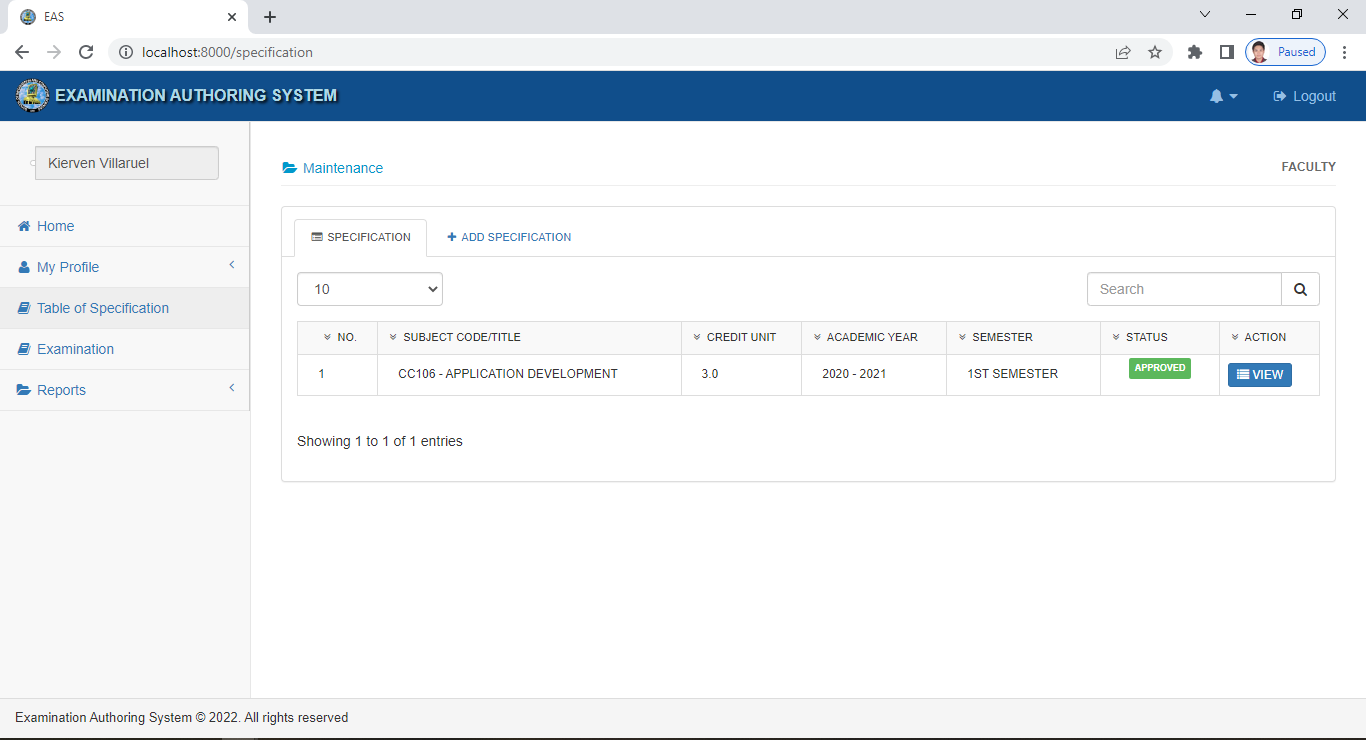
**Figure 4.9** *User Registration (Faculty and Program Head)*

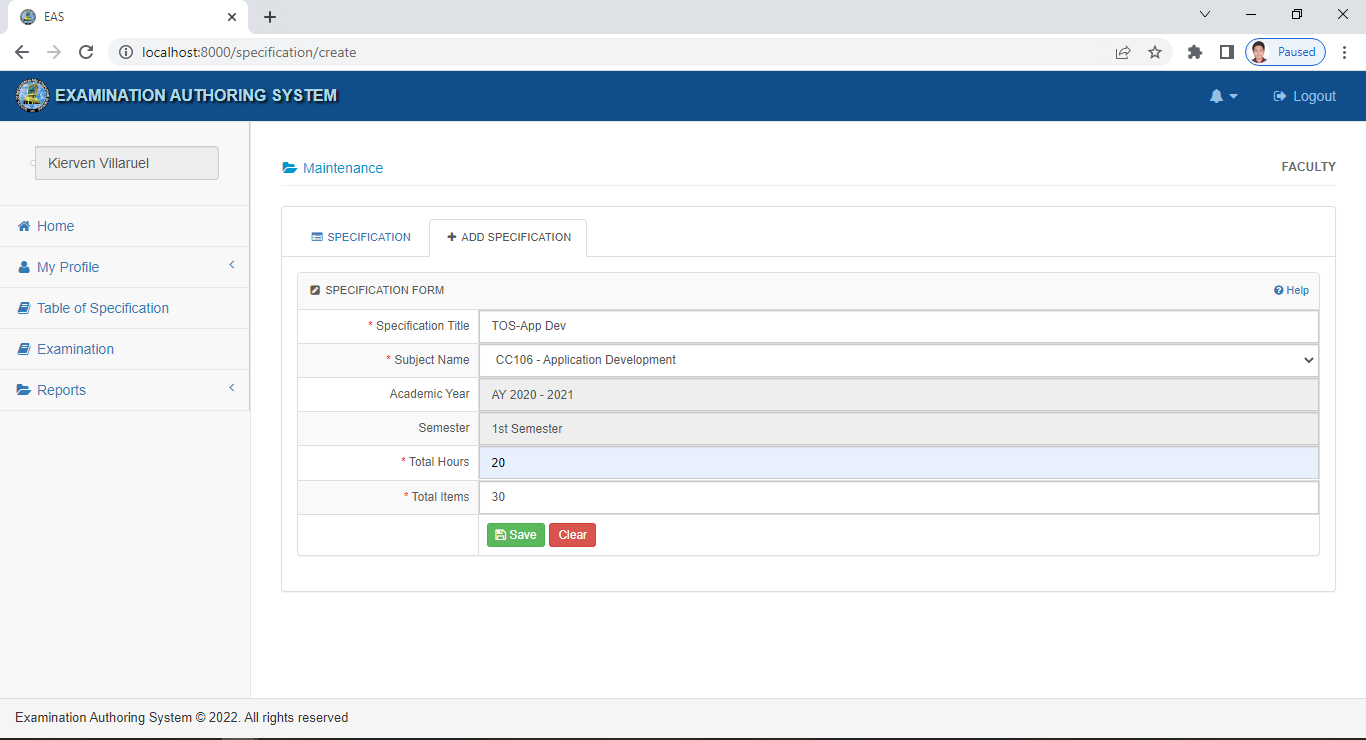
The System Administrator, specifically the Dean, is the one who has the privilege of registering the user in the system. She is the only one allowed to activate and deactivate a user account if necessary. In case there is a new administrator to take over, the current administrator should be deactivated, making one administrator active at a time.



**Figure 4.10** *Assigning of Subject to Faculty Members*

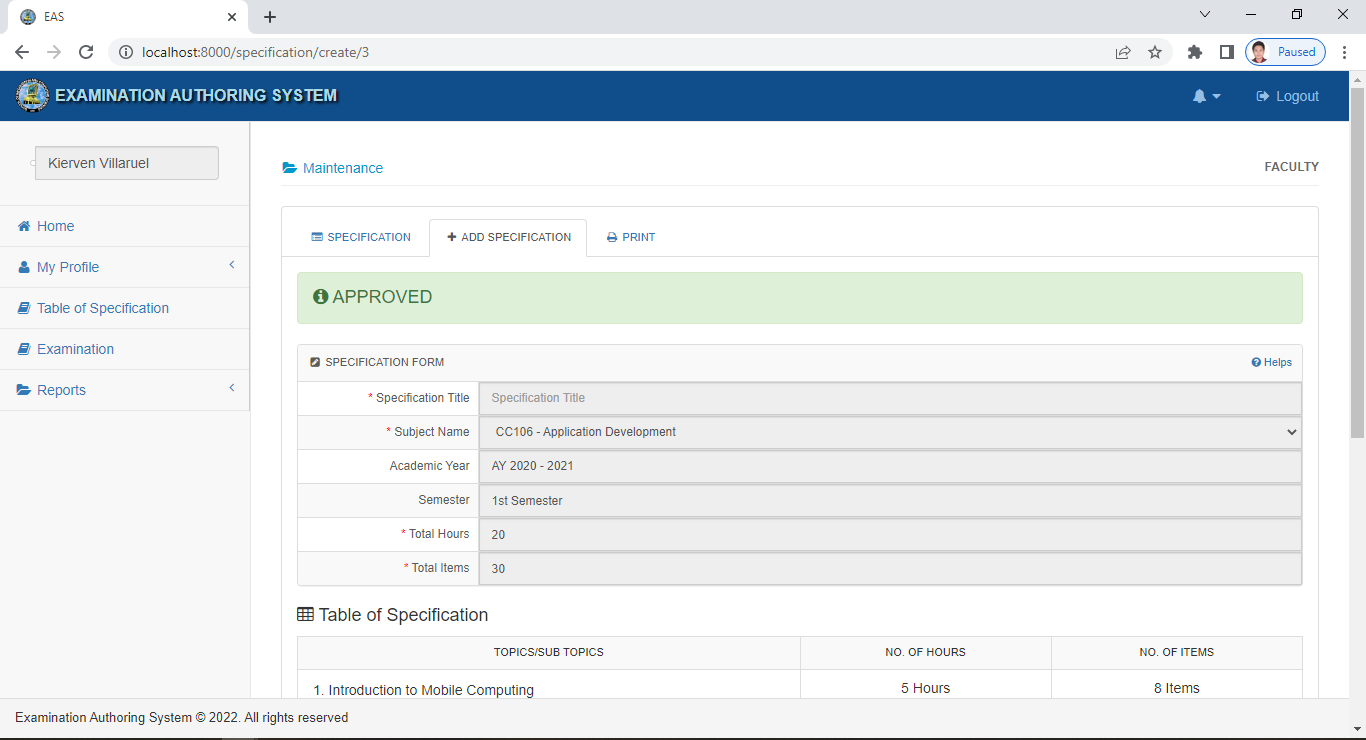
As shown in Figure 4.10, the administrator has the capability to assign subjects to faculty members. The administrator needs to click on the Assign Subject tab to access the form. The subjects that would be assigned to the faculty members were taken from the subjects active in the appropriate curriculum.

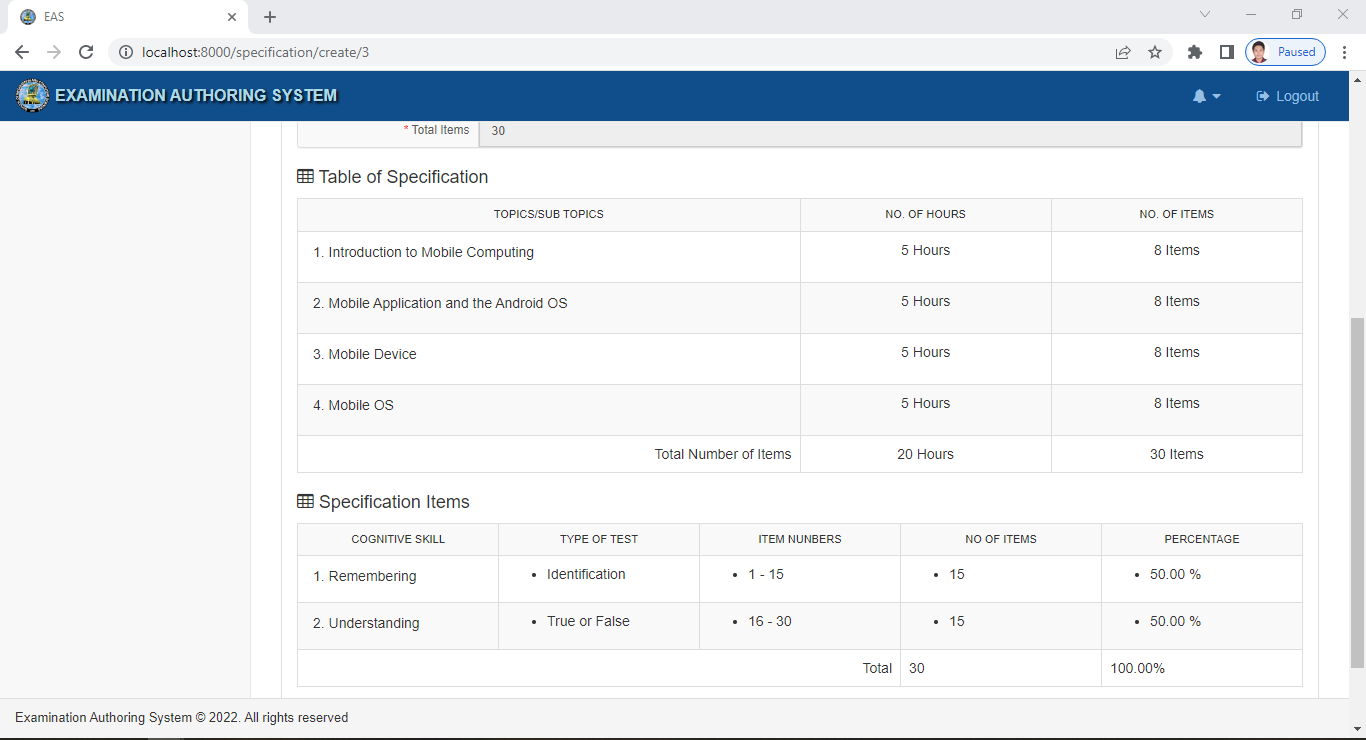




**Figure 4.11** *Create Table of Specification*

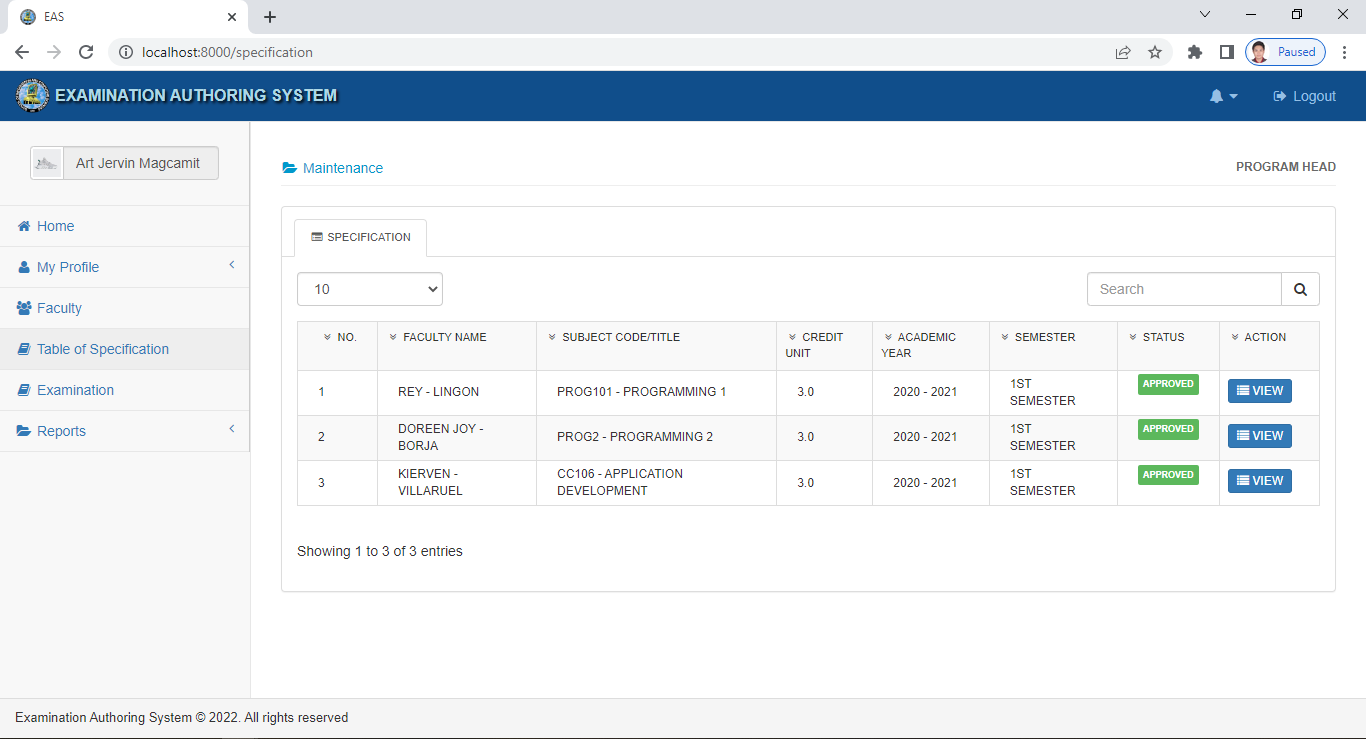
Before creating an exam, a table of specifications (TOS) should be made first. In figure 4.1, the faculty will create a TOS of the subject assigned to him considering the total number of hours the subject is taught, the type of examination, and the specified number of items. When done, the TOS will be submitted to the account of the Program Head for approval.





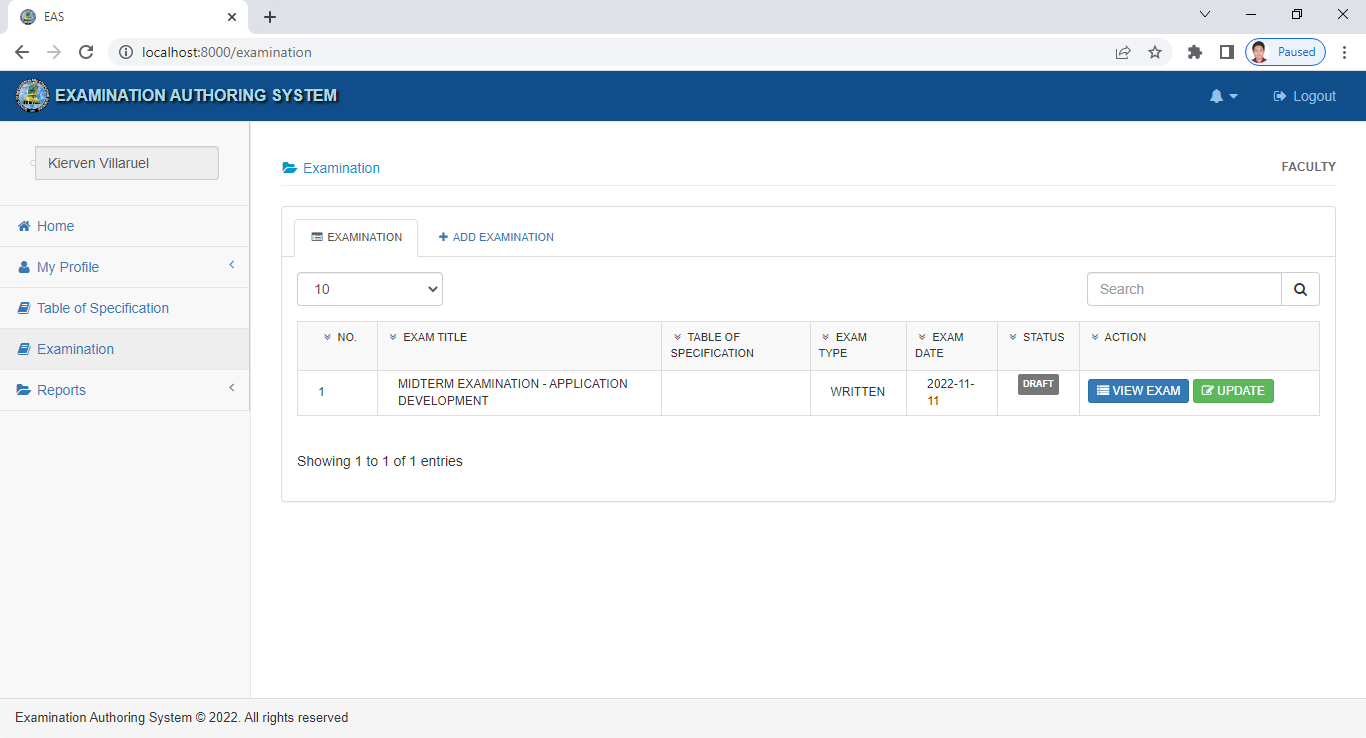
**Figure 4.12** *Competed and Approved Table of Specification*

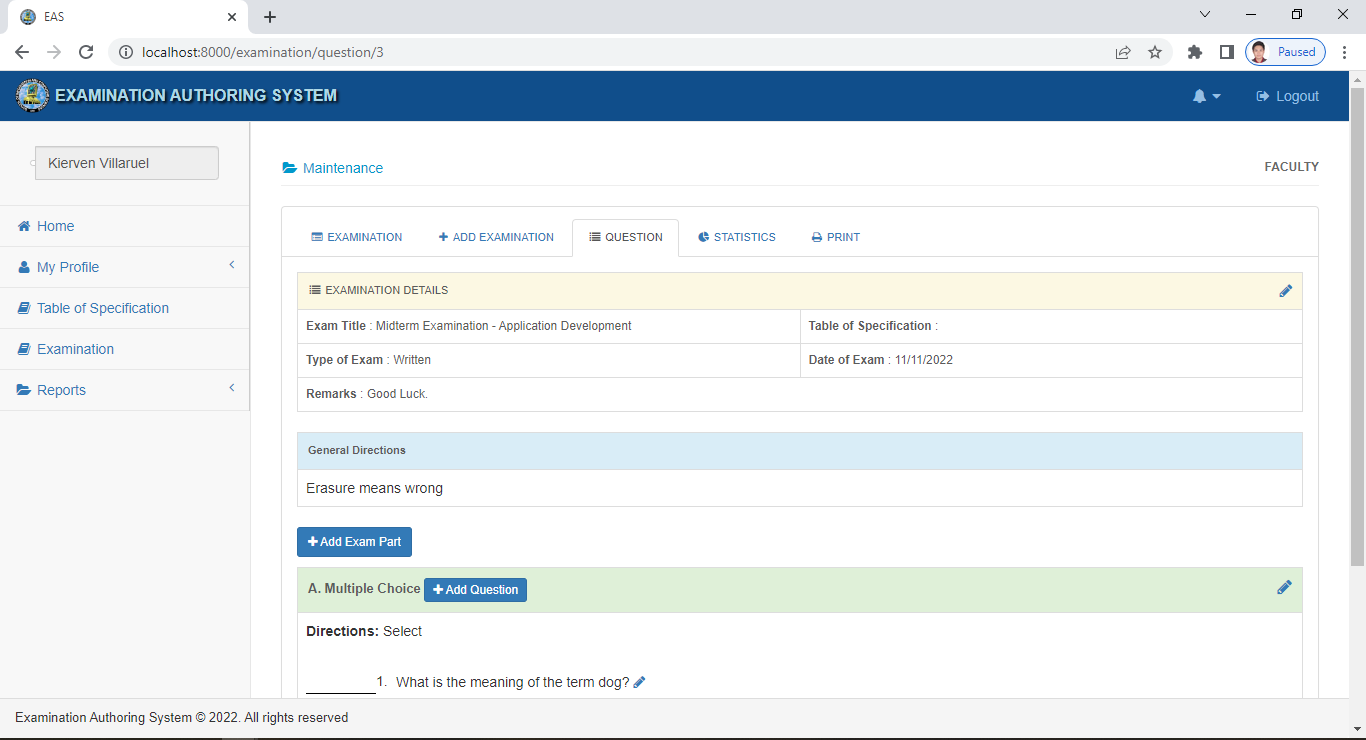
Figure 4.12 shows the completed and approved TOS for a specific subject. The created TOS was laid-out based on the format that is currently being used by the faculty member of IIST in creating their TOS. As shown it provides the topics details included in the examination that would be created for a subject. It also provide specific items of all the type of tests arranged per cognitive level.



**Figure 4.13** *Monitoring of submitted Table of Specification*

In the above figure, the Program Head can monitor the submitted Table of Specification by the faculty, same with the submitted examination as shown in figure 4.15. There are three statuses that can be seen in this monitoring page. TOS can have a draft status, when the faculty is not yet finished in creating his/her examination. Submitted status can be seen when the faculty member has completed and submitted the TOS.

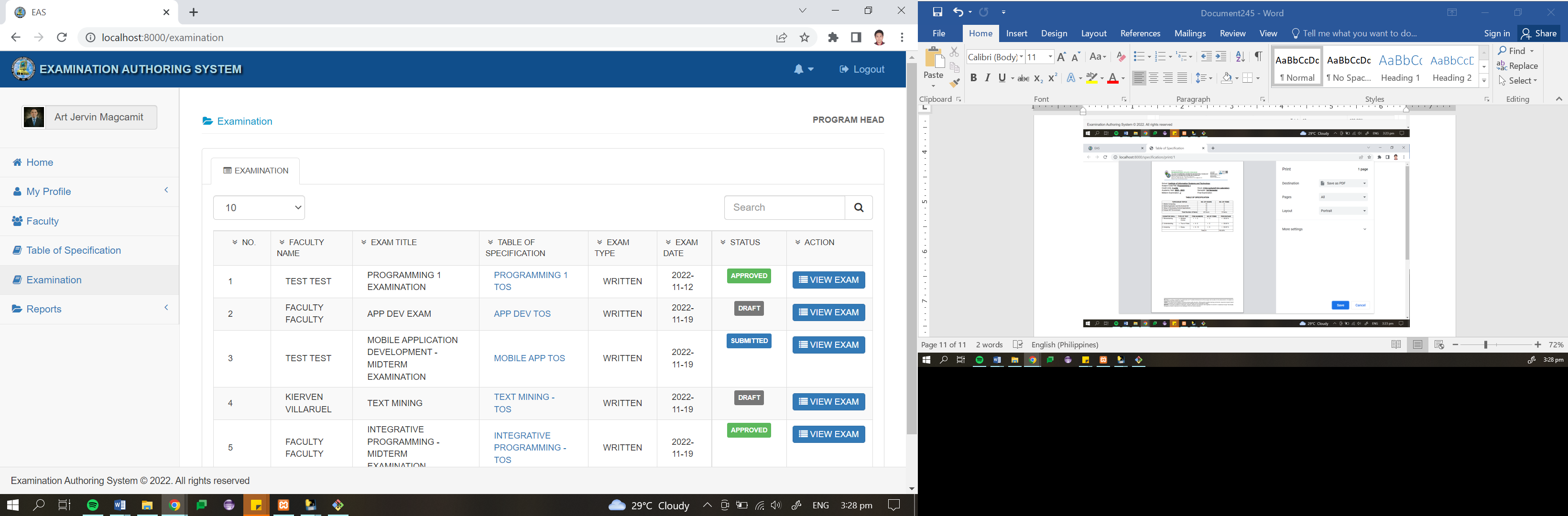




**Figure 4.14** *Creating of Examination*

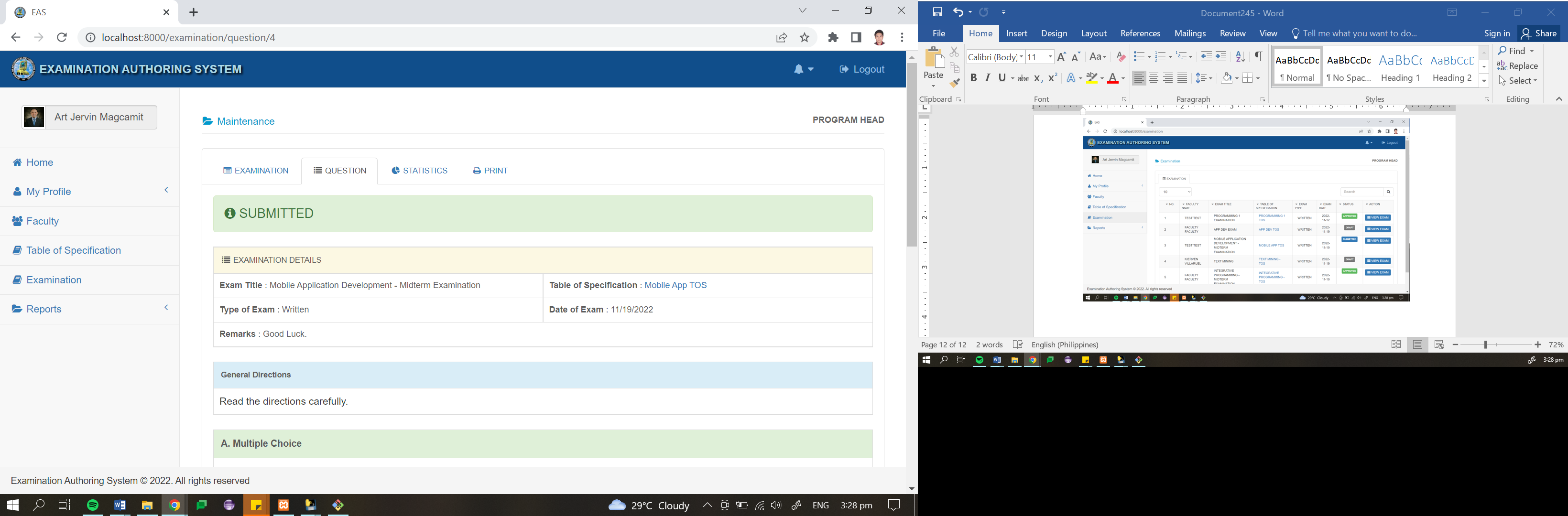
The main function of this examination authoring system is to create an exam and classify them according to their cognitive level using Bloom’s taxonomy. In Figure 4.14, the faculty is allowed to add questions based on their chosen examination type. The system will be the one to classify each question according to their cognitive level.

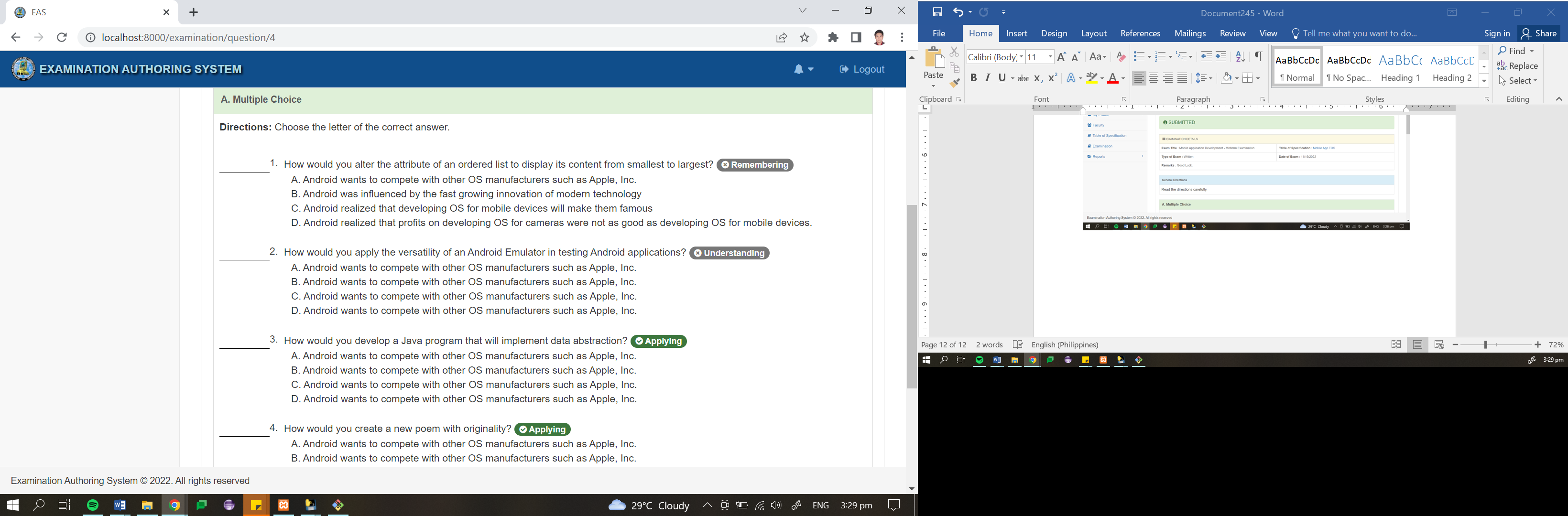
Once the examination is done, it will be submitted to the account of the program head for review and approval. The exam will be counter-checked to the submitted TOS.



**Figure 4.15** *Monitoring of submitted Examination*

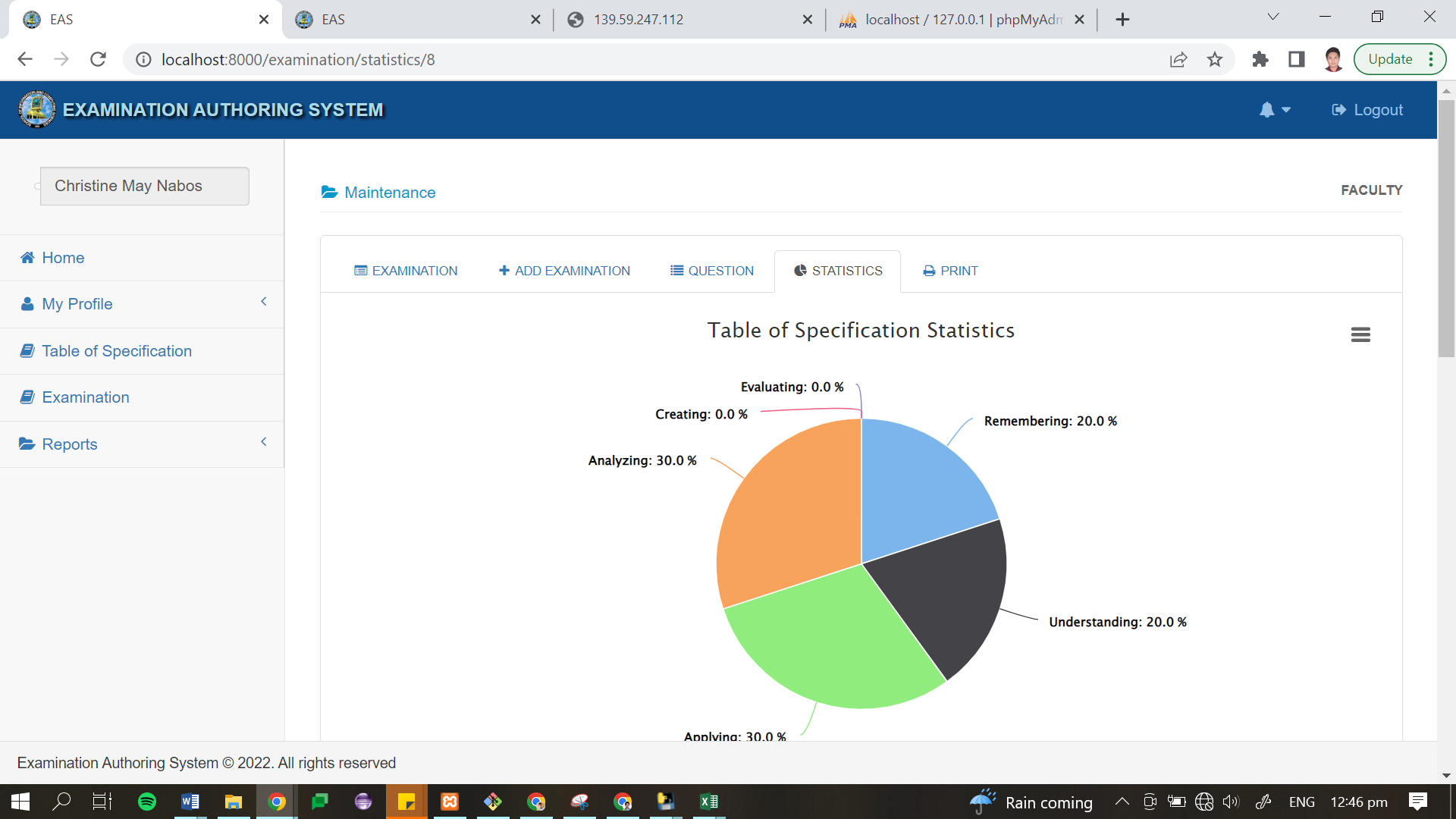
In the above figure, the Program Head can monitor the submitted Examination by the faculty, same with the submitted TOS as shown in figure 4.13. There are three statuses that can be seen in this monitoring page. Examination can have a draft status, when the faculty is not yet finished in creating his/her examination. Submitted status can be seen when the faculty member has completed and submitted the exam.





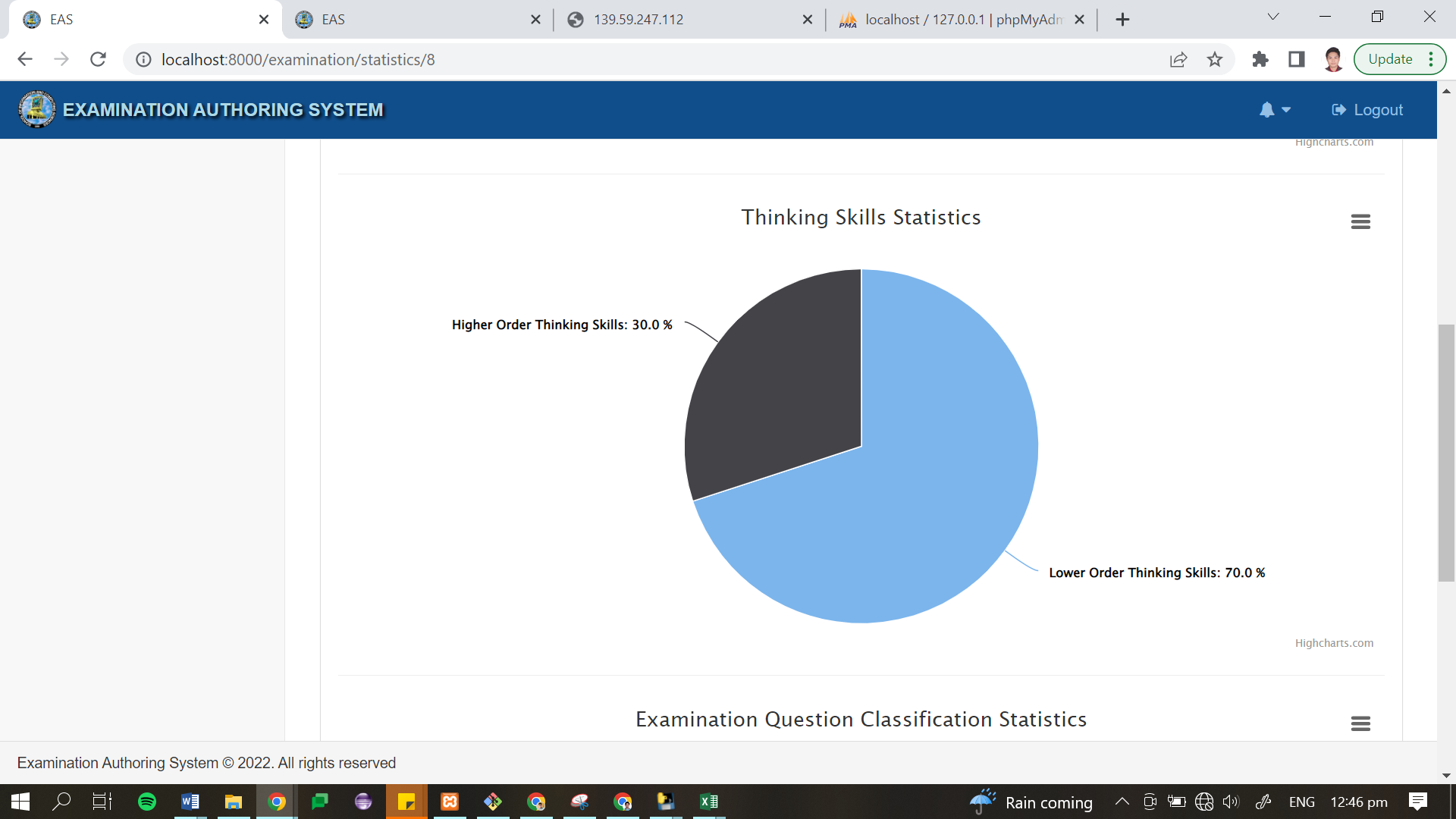
**Figure 4.16** *Completed Examination*

The images on Figure 4.16 shows the webpage that contains the completed examination of a specific subject. As presented, it contains the details of the subject and the questions per type of tests with its matching cognitive level as automatically classified by the system. The classifications were provided based on the implemented NLP and the application of the Naïve Bayes Classifier.



**Figure 4.17** *Exam Question Classification Report*

Figure 4.18 shows the examination questions classification report. This report was automatically generated from the summary of the questions per examination with their corresponding cognitive classification on Bloom’s Taxonomy.



**Figure 4.18** *Examination Cognitive Skills Report*

As shown in figure 4.18, it provides the overall summary of the examination question with regards to the Cognitive skills that the exam can assess. The representation includes the percentage of questions for the Higher Order Thinking skills and the Lower Order Thinking skills. This kind of report can help faculty members to assess the overall quality of their examination.

**Software and Hardware Specification**

**Table 4.15** *Minimum Hardware and Software Requirements of the system*

|  |  |
| --- | --- |
|  | **Specification** (Web Server) |
| Operating System | * Windows 8, Windows 10 or Higher |
| Special Software | * Composer v1.7.3 |
| Hardware | * Core-i5 Processor(CPU) with 2.30 gigahertz (GHz) frequency or above * A minimum of 4GB RAM * Monitor Resolution 1024x768 or higher * A minimum of 500GB available space on the hard disk * USB Keyboard and Mouse or some other compatible pointing device |
| Network | * Internet Connection Speed of at least 50MBps |
| Browsers | * Chrome 36+ or * Mozilla Firefox 31+ or * Internet Explorer 11+   *Browser Configuration:*   * JavaScript must be enabled * Cookies must be enabled * Pop-up windows must be enabled |

It can be seen in Table 4.15 the minimum required hardware, software and network specification of the examination authoring system. Since the system was designed with a Laravel framework basically to improve its program security, the minimum requirements are a little bit higher than the requirement of typical we-based systems. Moreover, special software, specifically the Composer is also needed to support some features of Laravel framework.

**Presentation of Testing Results**

Testing and evaluation procedures were conducted after the development of the final prototype. The developed examination authoring system underwent a cross-browser compatibility test. It was accessed using Google Chrome, Microsoft Edge, Internet Explorer, and Mozilla Firefox. Of the different browsers used, Google Chrome was the most compatible as the system did not experience any software malfunction in the web interface and system functionalities. Meanwhile, when tested with other browsers, like Internet Explorer, some pop-up menus and JavaScript prompts experienced irregularities due to outdated browser version and blocked web technology extensions. Stress testing was also conducted to verify the system’s robustness and error handling. This testing procedure determined possible error occurrences if invalid and out-of-range inputs were entered into the system. This helped the developer to plan accordingly the response that the system would display if such errors happened unexpectedly.

User Acceptance Testing was also conducted. The system was presented to the Associate Dean, Program Heads, and faculty members of the Institute of Information Systems and Technology. Pictures taken during the presentation are presented in Appendix K. Several faculty members tested the Examination Authoring System themselves. The system was made available via the local area network using XAMPPServer remote configuration. UAT forms were given to program heads and faculty members to gather their acceptance, views, comments, and recommendation when it comes to the functionalities of the system. Based on the result, most of the retrieved forms (see sample in Appendix I), gathered positive outcomes presented in Table 4.15, most of the faculty members strongly agreed that the system has executed all the functions that were tested. Hence, the acceptance of the system’s features and function are strongly agreeable. While some faculty members provided comments on some functionalities, the developer took these comments for the improvement of the functions of the whole system.

**Table 4.16** *User Acceptance Results*

|  |  |  |
| --- | --- | --- |
| **System Requirements** | **Mean** | **Adjectival Interpretation** |
| User Access Management | 4.5 | Very Strongly Agree |
| Creation of Table of Specification | 4.4 | Very Strongly Agree |
| Creation of Examination | 4.3 | Very Strongly Agree |
| Submission Monitoring | 4.7 | Very Strongly Agree |
| Report Visualization | 4.6 | Very Strongly Agree |
| **Grand Mean** | **4.5** | **Very Strongly Agree** |

Five (5) system requirements were evaluated by the actual users of the organization. 10 faculty members of MSC – IIST including the program heads, tested the system. As result, faculty members very strongly agree that the developed Examination Authoring system met the provided requirements, with a grand mean of 4.5. Hence, the acceptance of the system's features and function are strongly agreeable. While some faculty members provided comments on some functionalities, the developer took these comments for the improvement of the functions of the whole system.

After UAT, the system was also subject to quality evaluation. The developer used an evaluation form which sample is presented in Appendix I. The results of the conducted evaluation based on the ISO/IEC 25010 are detailed in Table 4.16.

**Table 4.17** *Software Quality Evaluation Result of the Developed System*

|  |  |  |
| --- | --- | --- |
| **System Attribute** | **Mean** | **Adjectival Interpretation** |
| Functional Suitability | 4.7 | Very Strongly Agree |
| Performance Efficiency | 4.8 | Very Strongly Agree |
| Compatibility | 4.7 | Very Strongly Agree |
| Usability | 4.7 | Very Strongly Agree |
| Reliability | 4.6 | Very Strongly Agree |
| Security | 4.6 | Very Strongly Agree |
| Maintainability | 4.8 | Very Strongly Agree |
| Portability | 4.8 | Very Strongly Agree |
| **Grand Mean** | **4.7** | **Very Strongly Agree** |

This table displays the overall result of the software evaluation. The testers evaluated the system with a mean of 4.7 for Functional Suitability, Compatibility, and Usability. The mean of 4.8 was given to Performance Efficiency, Maintainability, and Portability. On the other hand, Reliability and Security got the lowest mean of 4.6. This mean that the evaluators are in agreement with the overall quality of the system. The resulting grand mean of 4.7 is interpreted as “Very Strongly Agree”, which means that the system suits what is being expected by the respondents.

**Chapter 5**

**SUMMARY, CONCLUSION, AND RECOMMENDATION**

**Summary of Findings**

The learning process of students depends on what teachers ask them to do during class discussions or in taking periodical assessments. Administering major examination is way to assess the learning of students that they got from different subject matters. It is best that assessments like major examinations, which cover most of the percentage of their grade, are well created using the prescribed standards. To attain this goal, the Examination Authoring System was materialized as a product of this study. The purpose of this study is to develop a centralized web-based Examination Authoring System that serves as an authoring tool to create a Table of Specifications and Major Examination that classifies the questions in the examination according to the six cognitive levels of the revised Bloom’s Taxonomy.

It was found that machine learning and the application of natural language processing are helpful when dealing with text classification. Natural language processing is very powerful and is used in various ways that could help us attain a specific goal. Utilizing available technologies and programming languages is beneficial to people in the academe to make their tasks as easy as possible, like checking and monitoring document submissions. When using a classification algorithm to do the text classification process, it is evident that there should be plenty of data that needs to be accumulated to be used in training a specific classifier. In scenarios like creating questions for the right cognitive level, different and uncontrolled variables can lead the text classifier to have incorrect predictions. Thus, it is best to uncover and consider them to apply text classification in a larger context.

**Conclusion**

After conducting a series of tests, the system has proven its capability to improve the existing system used by the MSC – Institute of Information Systems and Technology. All objectives of the project have been met, including the creation of the table of specification and examination in the system wherein these two related documents are placed in one centralized platform. The authoring system also has the capability to monitor submission and checking of TOS and examination. The best thing about this system, is that it can classify questions on created examinations based on the cognitive level of the revised Bloom’s Taxonomy, which is a significant feature that can eliminate problems experienced by faculty members in making good examinations. And lastly, it is a great help for faculty members to see the statistics of their examinations through graphical representation that will enable them to have a better outlook in providing better assessment tools for their students.

Therefore, it is concluded that the developed system would be a great help to the faculty and staff of MSC – IIST in upgrading their ways on how to do things and in solving encountered problems.

**Recommendation**

Having the summary of findings and conclusion as basis, the following recommendations were drawn:

1. Upon using the system, make sure to have the required minimum hardware and software requirements.
2. The exam authoring system would function better if uploaded to a dedicated server.
3. The system must be prepared to be connected with the other modules of a bigger system to serve a better purpose.
4. Faculty members should be familiar with the Art of Questioning.
5. It also recommended trying to improve the NLP process with a more profound and high-level approach.
6. Other classification algorithms like Support Vector Machines and Voting Algorithm may also be used to compare the accuracy in classifying examination questions according to the revised Bloom’s Taxonomy.
7. Plenty and diverse collection of data must be used in training the classifier.
8. Lastly, it is recommended to test the system with a bigger number of possible users to measure the full potential of the question classification.