

A Comprehensive Examination Assessment Model using Machine Learning

C K Marigowda

Department of Information Science and Engineering
Acharya Institute of Technology
Bengaluru, India

Anshula Ranjit

Department of Information Science and Engineering
Acharya Institute of Technology
Bengaluru, India

Ashwin Prasad A

Department of Information Science and Engineering
Acharya Institute of Technology
Bengaluru, India

Bharath Karanth

Department of Information Science and Engineering
Acharya Institute of Technology
Bengaluru, India

Sarthak Sharma

Department of Information Science and Engineering
Acharya Institute of Technology
Bengaluru, India

Abstract-The main objective of this work is to develop an application for educational institutes. The app will be foremost required by any students and professors to ease the examination process and experience. Using Item Response Theory (IRT), our system can tailor a unique testing experience for every test-taker based on his ability, while the system evaluates the answer responses 'on the fly' using various Natural Language Processing (NLP) techniques and a probabilistic classifier. With the help of this system and human evaluators, a fair analysis of the test-takers ability can be inferred. With the use of Machine Learning algorithms, our proposed system evaluates the answers provided with the utmost accuracy, for assessments computers provide quite a few advantages for both faculties and test-takers. Nonetheless, in literature, there is no agreement on the equality of paper-and-pencil and computer-based test environments. The system expects each subject to be divided into multiple modules. The test-taker will be tested for each module separately. The cumulative scores for each of these individually tested modules will give the score for the subject.

Keywords: *Item Response Theory, Machine Learning, Computer-Based Test, Artificial Neural Networks, NLP.*

I. INTRODUCTION

In the field of education, Assessing student ability plays a vital role in all levels. The assessment allows individual schools to plan their improvement in working with students. It also facilitates college admissions committees grade individual student performance and aptitudes [1]. Examinations are a significant tool for student assessment. As stated by Brown, Race, and Bull the methods of assessment can have a significant impact on student learning. It has been suggested that if an aspect of a course is not assessed, students will probably not learn it. Hence, assessing students' performance is always a significant issue for educational systems. Depending on the number of students, it is not easy to implement such techniques more often. In that sense, Computer-Based Testing (CBT) systems can offer

alternatives for implementing tests more often in different educational settings [2].

To approve and validate computer-based testing over more customary traditional paper-and-pencil testing, in any case, guarantee that distinction in the test doesn't adversely affect students' performance.

For assessments computers provide quite a few advantages for both faculties and test-takers. Nonetheless, in literature, there is no agreement on the equality of paper-and-pencil and computer-based test environments.

II. OBJECTIVE OF THE PROPOSED SYSTEM

The objective of the proposed system is to provide a fair way of evaluating every test taker based on his skill level, rather than the current way of linearly testing everyone who takes the same test. This way, we believe that we can provide a much better understanding of the test-takers' skill level and in turn, help the test-taker improve the areas of lack.

Another objective of this system is to reduce the latency between when the test is taken and when the evaluator gets to evaluate the responses. With the help of the cloud, this can be done in near real-time, evaluating every test right after it's complete. This also helps in reducing the carbon footprint as the system is entirely digital.

A. Proposed System

The system consists of client software which is presented to the test-taker. On the other end, the faculties will also have a portal to submit their questions along with the features of the question i.e. Answer key, keywords, key sentences, question specific things along with the trait function, and the unique score multiplier.

This system dynamically analyses and evaluates the answer responses from the test-taker and awards temporary scores. These scores will be used to present the questions which come later on.

These temporary scores are only used to dynamically change the questions being presented based on the skillset of the test-taker. Thus, the answer responses need to be evaluated physically by a human evaluator, which is then multiplied with the score multiplier for the question, to derive the final score of the test-taker. It deals with tagging every individual document with its sentiment. In Document level, the whole document is either classified into positive or negative class.

B. Advantages of the proposed system

- A fair and more accurate evaluation of the skill level of the test-taker.
- A better understanding of one's areas of expertise and areas of lack.
- A completely digital system, which reduces the use of papers significantly.
- Near real-time evaluation considering the latency reductions between the evaluator and the test-taker, compared to current methods.
- An intelligent system which predicts the ability of the test-taker as he is responding to questions.

III. SYSTEM ANALYSIS

A. Functional Requirements

It manages the functionalities required from the system which are as per the following:

This application will assist universities/colleges to tailor a unique testing experience to every test-taker.

- The only authorized person, faculty can prepare a test or students can take the test.
- Faculty can set question papers for respective modules in every subject.
- Every question is should be awarded difficulty grade to differ from other questions.

B. Non-Functional Requirements

There are some non-functional requirements such as:

- **Performance:** The number of client-side machines that can be supported at once, depends on the server that the institute or the educational body wants to opt for. Regardless, the web-application server should provide great performance with techniques such as in-memory caching, and localized databases for quicker data-fetching. Upon completion, the data of the student's examination performance should be calculated in the order of seconds before being sent to the teacher for a physical evaluation.
- **Availability:** The examination system should be available when needed. As examinations are not day-to-day occurrences/events, the system-server availability should be optimized for when the examinations are planned. During this period, however, the system should have UPS backup to ensure 100% uptime, up and till the point all the data needed, is either download to an offline system or is processed by external evaluation software.
- **Reliability:** Reliability is the extent to which a program performs with the required precision. The reliability of the application needs to be high for the reasons that it deals with valuable student's data. Security is also a high priority as data collected here is considered to be sensitive and precious. On the teacher's end, questions

need to be treated with the utmost security as leaking this would nullify the validity of the examination it was a part of.

- **Usability:** The usability of the application must be very high, as the target audience for the app is generally in the older age-group. Services like session management and authentication persistence should be at the core of the application
- **Portability:** As the application will either be a desktop application written in ElectronJS or a Web Application based on the needs of the educational body, it should offer extreme portability. ElectronJS can be complied with Windows, MacOS, and Linux and the Web is platform-independent.
- **Flexibility:** As a modular application, flexibility should come at no extra cost. The application is based on 4 modules at its core and these modules are in turn extremely modular. Changes done in one module should not affect the changes in the other modules as the API Spec for these should be invariant of these changes. Furthermore, employing the micro-service architecture, the application should provide flexibility and scalability at a very low cost.

C. Specific Requirements

Since students and faculty are the primary objective gathering of our site, we will only concern about some important functions for the user.

Students:

- The student is provided with USN and password and login to the system.
- After successfully logging to the system student is presented with a baseline test.
- After clicking on the start button, the exam starts along with the timer.
- The baseline test is a pool of questions with different levels of difficulty.
- Based on the performance of the baseline test, a student is presented with a question with a level of difficulty.
- The level of difficulty hikes/drops based on the performance of the previous question.
- A student can end the test by clicking the 'submit' button, which can only be done by answering all the rounds of questions.

Faculty:

- Faculty are provided with username and password and log into the system.
- After a successful login, the faculty are provided with the option, to select the test name.
- Under the test name, we have a list of students who have attended this test.
- On selecting a student, we have answers sheet of the respective student, for each question, our system evaluates the answer with a grade.
- Faculty evaluate the answer script manually and be prompted to send the feedback for each question, to compare the grade awarded to the question is correct or not.

- After evaluating the answer script, by clicking the 'submit' button will redirect back to the list of students for evaluating other scripts.

D. Software Interface Requirements

It consists of a dashboard having the option of login for faculty and students and after logging in Feedback option provided for each question for the faculty side.

We are using the following technologies:

Front-end: The UI is built using the javascript library called React.

Back-end: The backend is built on Electronjs which is an open-source software framework that uses Node as a library. It is used to desktop applications with HTML5, CSS, and JavaScript.

Database: NoSQL is a non-relational database that does not need defining its structure, unlike MySQL.

- E. Python was found by Guido Van Rossum in Netherland,
F. 1989 which has been public in 1991[31]. Python is a
G. programming language that's available and solves a computer
H. problem which is providing a simple way to write out a
I. solution [31]. [32] mentioned that Python can be called as a
J. scripting language. Moreover, [32] and [32] also supported
K. that actually, Python is a just description of language because it
L. can be one written and run on many platforms. In addition,
M. [34] mentioned that Python is a language that is great for
N. writing a prototype because Python is less time consuming and
O. working prototype provided, contrast with other programming
P. languages.
Q. Many researchers have been saying that Python is efficient,
R. especially for a complex project, as [33] has mentioned that
S. Python is suitable to start up social networks or media
T. steaming projects which most always are a web-based which is
U. driving a big data. [34] gave the reason that because Python
V. can handle and manage the memory used. Besides Python
W. creates a generator that allows an iterative process of things,
X. one item at a time and allow program to grab source data one
Y. item at a time to pass each through the full processing chain Python was found by Guido Van Rossum in Natherland, 1989 which has been public in 1991[31]. Python is a programming language that's available and solves a computer problem which is providing a simple way to write out a solution [31]. [32] mentioned that Python can be called as a scripting language. Moreover, [32] and [32] also supported that actually Python is a just description of language because it can be one written and

run on many platforms. In addition, [34] mentioned that Python is a language that is great for writing a prototype because Python is less time consuming and working prototype provided, contrast with other programming languages. Many researchers have been saying that Python is efficient, especially for a complex project, as [33] has mentioned that Python is suitable to start up social networks or media steaming projects which most always are a web-based which is driving a big data. [34] gave the reason that because Python can handle and manage the memory used. Besides Python creates a generator that allows an iterative process of things, one item at a time and allow program to grab source data one item at a time to pass each through the full processing chain.

- Z. Python was found by Guido Van Rossum in Natherland,
AA. 1989 which has been public in 1991[31]. Python is a
BB. programming language that's available and solves a computer
CC. problem which is providing a simple way to write out a
DD. solution [31]. [32] mentioned that Python can be called as a
EE. scripting language. Moreover, [32] and [32] also supported
FF. that actually Python is a just description of language because it
GG. can be one written and run on many platforms. In addition,
HH. [34] mentioned that Python is a language that is great for
II. writing a prototype because Python is less time consuming and
JJ. working prototype provided, contrast with other programming
KK. languages.
LL. Many researchers have been saying that Python is efficient,
MM. especially for a complex project, as [33] has mentioned that
NN. Python is suitable to start up social networks or media
OO. steaming projects which most always are a web-based which is
PP. driving a big data. [34] gave the reason that because Python
QQ. can handle and manage the memory used. Besides Python
RR. creates a generator that allows an iterative process of things,
SS. one item at a time and allow program to grab source data one
TT. item at a time to pass each through the full processing chain.
UU. Python was found by Guido Van Rossum in Natherland,
VV. 1989 which has been public in 1991[31]. Python is a
WW. programming language that's available and solves a computer
XX. problem which is providing a simple way to write out a
YY. solution [31]. [32] mentioned that Python can be called as a

ZZ. scripting language. Moreover, [32] and [32] also supported
 AAA. that actually Python is a just description of language because it
 BBB. can be one written and run on many platforms. In addition,
 CCC. [34] mentioned that Python is a language that is great for
 DDD. writing a prototype because Python is less time consuming and
 EEE. working prototype provided, contrast with other programming
 FFF. languages.
 GGG. Many researchers have been saying that Python is efficient,
 HHH. especially for a complex project, as [33] has mentioned that
 III. Python is suitable to start up social networks or media
 JJJ. steaming projects which most always are a web-based which is
 KKK. driving a big data. [34] gave the reason that because Python
 LLL. can handle and manage the memory used. Besides Python
 MMM. creates a generator that allows an iterative process of things,
 NNN. one item at a time and allow program to grab source data one
 OOO. item at a time to pass each through the full processing chain.

IV. SYSTEM DESIGN AND METHODOLOGY

This section provide the insight to the software design phase. While the requirement specification activity deals entirely with the problem domain [3]. In the design phase, the client and business necessities and specialized contemplations all come together to formulate a product or a system.

A. Methodology of the proposed system

The system will be implemented as a web application or an android application which will be presented to the test-taker on a tablet. The first question presented to the test-taker will be neutral in nature and will be used solely to gauge the test taker's ability baseline. His response to the first question sets the precedence for the questions that will follow. This response, like every other, will be processed by a Machine Learning model that is trained to accept features from the test-takers response, along with the features of the question that the response is pertaining to. These features include keywords, key sentences, and other question specific things. The score given out by this model will be used to predict the next question which will be presented to the test-taker. This is decided based on the ability curve of the question. Thus, only questions that have the highest likelihood of being answered by the test-taker, will be presented to him. If his ability is low, then the score multiplier for that question will below. However, if his ability is predicted to be high, the question will be tough, and the score multiplier will also be high.

This way, two test-taker with opposite skill levels, are expected to answer all the questions presented, albeit their scores will differ thanks to the score multipliers.

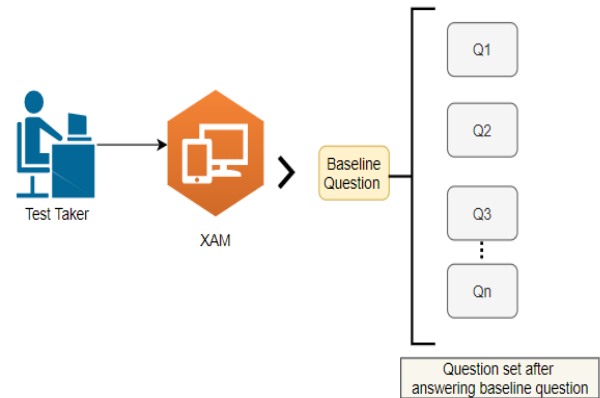


Fig. 1. Overview of Test Taking Process

B. High-Level Design

High-level design (HLD) explains the architectural design that may be used for developing software products [7]. The design diagram provides an outline of a whole system, distinguishing the fundamental components most elements that need to be developed for the product and their interfaces. The high-level design provides a view of the system at an abstract level. It shows how the major pieces of the finished application will fit together and interact with each other [8]. In this design, we can see witness the process overview where the Actor who is the test-taker attempts a baseline test to determine its ability level and henceforth in adaptive questioning round the real assessment begins and evaluation is done for the answers submitted by the test-takers. Based on the evaluation mark provided by the system the next set of questions is provided to the test-taker.

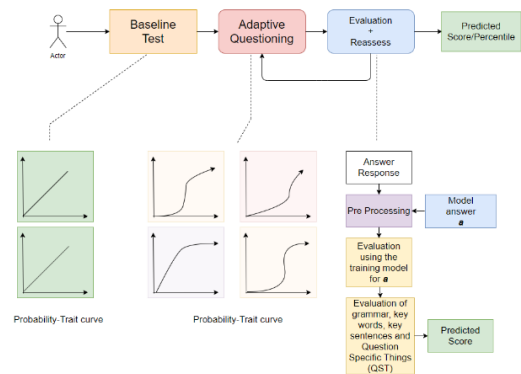


Fig. 2. High-Level Design

Steps involved in the process of system design:

- Establishing the baseline
- Selecting the question to be presented based on the baseline
- Evaluate the new baseline and repeat

C. Module Based Design

The system expects each subject to be divided into multiple modules. The test-taker will be tested for each module separately. The cumulative scores for each of these individually tested modules will give the score for the subject.

Each module has a baseline test, which determines the skill level for the upcoming questions. After the baseline test, his performance dictates the difficulty of the next question shown. This again is analyzed and evaluated by the machine learning algorithm and the cycle continues. Eventually, after completing responding for all questions in the module, the responses are sent to the human evaluator along with the score multipliers and the system prepares for the next module. This continues until all modules in that subject are tested for. After this, the average score secured in all the modules will be the final score for the subject.

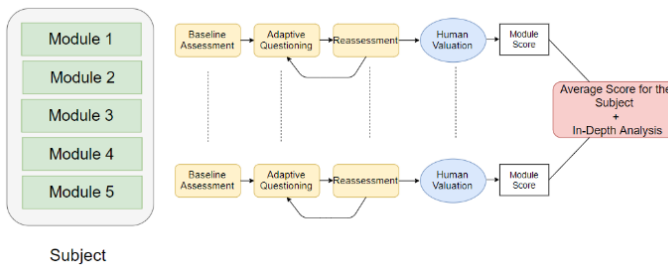


Fig. 3. Module Based Design

D. Modules of the Proposed System

The proposed system consists of the following modules that make up the system.

- User Interface Module
- Backend Module
- Item Response Theory Module
- Grading Module

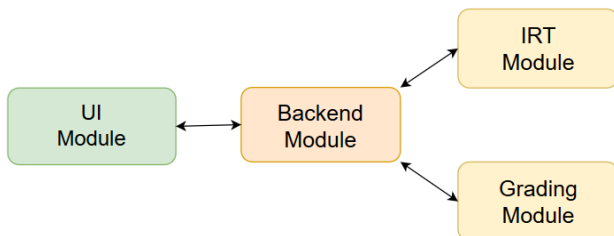


Fig. 4. Modules of the System

E. Workflow of the UI System

The workflow of our system is straightforward; here the actor is anybody who is taking up the test. The system is a desktop application upon clicking the icon; the actor is greeted by a login screen where they need to enter their correct credentials. For a first-time user sign up option is also available. Once login is successful actor can see the different tests allotted to the candidate, clicking on the correct test different subjects are shown and the actor can choose the subject they have to appear for, the test is activated at the correct time and then the actor can start taking the test by clicking start test. At the start of the test, few base test questions are provided by the system based on the score a set of questions with ranging difficulty level is provided. If the test is completed then you go back to the test subjects' page, if not, the questions are provided with lesser difficulty level until the actor is able to complete the exam.

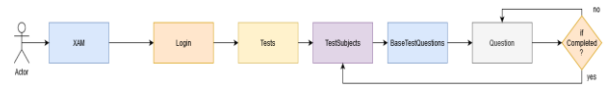


Fig. 5. Workflow of the User Interface

F. Backend UI Integration

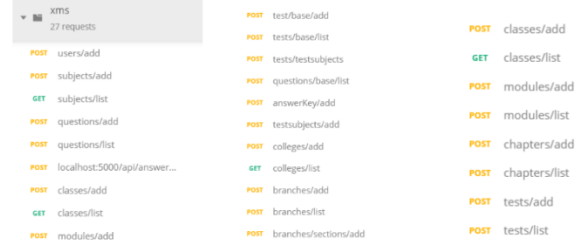


Fig. 6. Backend User Interface Integration

- /api/users
 POST /api/users/add
 Adds user's details (name, email, password, role, branch, section) to database with proper validation and sent with a jsonwebtoken with user id as payload as a response.
- /api/auth
 GET /api/auth
 Get the user details from the database with correct authentication.
 POST /api/auth
 Send email and password to the above route, if email and password are valid which are equal as stored in the database, and sent with a JSON web token with user id as payload as a response. Otherwise error message of 'Invalid Credentials' is sent as a response.
- /api/tests
 POST /api/tests/add
 Adds test details of fields (name, college_id) to the database.
 GET /api/tests/list
 Get all the test details from the database with given branch_id.
 POST api/tests/testsubjects/add
 Add a Test Subject data with fields (subject_id, branch_id, section_id, base_id, test_id, module_ids, totalrounds) to the database. The 'totalrounds' field indicates no of questions should be provided to the given Test Subject.
 POST api/tests/testsubjects/list
 Get all the test subjects list from the database with given test_id, branch_id, and section_id.
 POST api/tests/base/add
 Add a Base Test with fields (question_ids, priority) to the database. The question_ids indicate the array of question_id which is references to the Question model where we have all sets of questions.
- /api/questions
 POST /api/questions/add
 Add a question with fields (chapter_id, question, rank) to the database. The 'rank' field determines the difficulty level of the question.
 GET /api/question/list
 Get the list of all questions according to given chapter_id as a parameter.
 POST /api/question/baseQuestions/list
 Get the list of all base questions with given base_id, user_id, and test_id.

POST /api/question/nextQuestion

Get the question for the current question round with given user_id and test_id. If the current round is equal to total round, it will send a response of completed: true in json.

- /api/answers

POST /api/answers/add

Add an answer to the database with respect to the question and the user.

- /api/branches

POST /api/branches/add

Adds a branch with fields (name, short_name, college_id) to the database.

POST /api/branches/list

Get the list of all branches with the given college_id.

POST /api/branches/sections/add

Adds section with field (name) to the database.

POST /api/branches/sections/list

Gets the list of sections from the database.

- /api/colleges

POST /api/colleges/add

Adds a college with fields (name, short_name) to the database.

POST /api/colleges/list

Get the list of all colleges from the database.

G. Database Design (NoSQL)

Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model. Database design involves classifying data and identifying interrelationships.

NoSQL database data model techniques include: Denormalization puts all data needed to answer a query in one place, typically a single database table, instead of splitting the data into multiple tables. Aggregates use light or no validation of data types, for example, strings or integers.

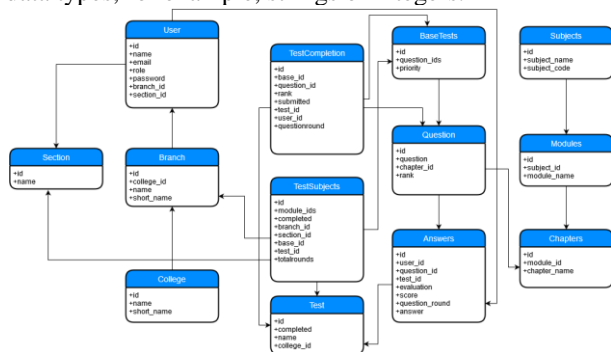


Fig. 7. Database Design

H. Item Response Theory

This theory talks about how the performance in a test depends on the items (questions) on the test.

In psychometrics, item response theory (IRT) (also known as latent trait theory, strong true score theory, or modern mental test theory) is a paradigm for the design, analysis, and scoring of tests, questionnaires, and similar instruments measuring abilities, attitudes, or other variables. It is a hypothesis of testing depends on the relationship between individuals' performance on a test item and the test takers' levels of

performance on an overall measure of the ability that item was designed to measure [4].

$$p_i(\theta) = c_i + \frac{1 - c_i}{1 + e^{-a_i(\theta - b_i)}}$$

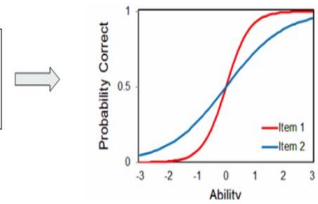


Fig. 8. Probability Correct vs Ability Graph

Where θ represents person abilities from a normal distribution which is used for estimating the item parameters. a_i , b_i and c_i are the item parameters. These determine the shape of the Item Response Function. Fig 9 depicts an ideal 3PL ICC.

Implementation of IRT

Three Steps:

- Establishing the baseline
- Selecting the question to be presented based on the baseline
- Evaluate the new baseline and repeat

o Establishing the Baseline

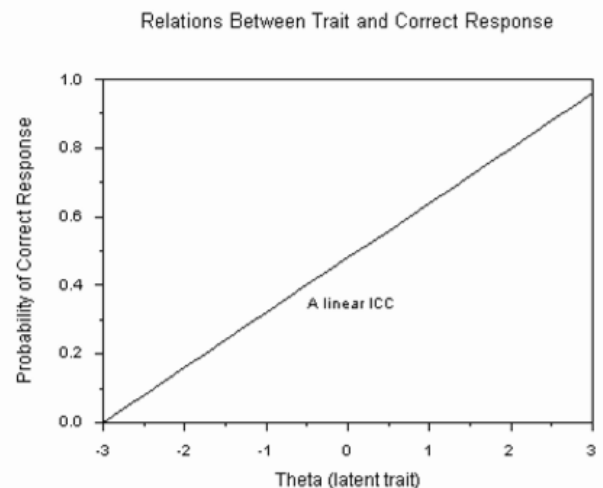


Fig. 9. Establishing the Baseline

- Starting point that defined step 1 of the process of selecting the trait value for selecting the upcoming set of questions to be answered
- Uses a linear curve to find the trait of the student.21.
- o Selecting the Question to be Presented Based on the Baseline

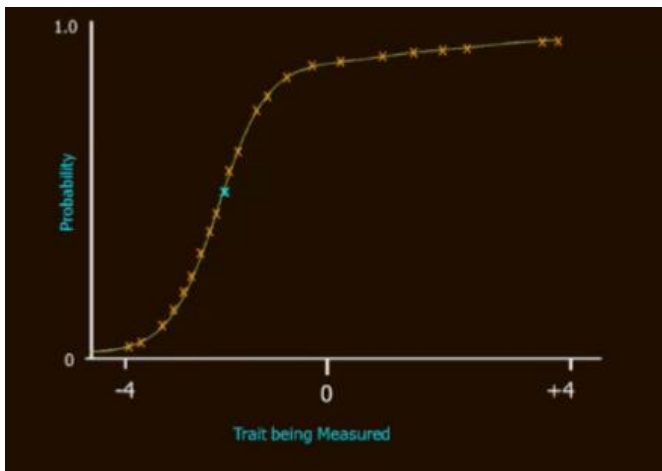


Fig. 10. Selecting Baseline Question

- Now we have the baseline ability of the test-taker. • With this, we can find the question with the most probability of being answered.

o **Evaluate the New Baseline and Repeat**

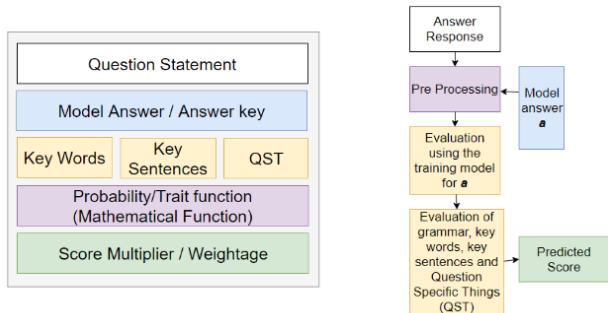


Fig. 11. Evaluating and Accessing Baseline

The machine learning pipeline for the system is initiated with the answer response. The pipeline then extracts the required features from the answer and then uses the provided answer key (model answer) to predict a score. This is also in consideration along with the keywords, key sentences, and other question specific things which are provided by the question setter. These aides the algorithm to provide a score between 0 – 10. These scores are then indicative of the current skill of the test-taker which is again used to provide an apt question as to the next one.

The predicted score is only temporary as the human evaluator also these responses, to provide a more realistic score to the test-taker. However, the difficulty of the question plays a huge role in the scores obtained as the score multiplier is inversely proportional to the easiness of the question, which is again decided on the skill level of the test-taker.

I. UML Sequence Diagram

UML sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes [5].



Fig. 12. Database Design



Fig. 13. Database Design

Figure 13 and 14 shows us the process of how the student interacts with the system for taking up the examination and hoe the faculty interacts with the system respectively.

J. Grading Design

This module is the last and final module that deals with the evaluation of the responses to the questions answered during the examination.

This is also known as the evaluation module. Evaluation intends to lead the assessment and to give marks or grades to students. Evaluation offers a way to determine whether an initiative has been worthwhile in terms of delivering what was intended and expected [6].

This involves six steps:

1. Data Collection for Text Analysis
2. Rubric Development
3. Text Analysis
4. Apply Machine Learning Algorithms
5. Classifying the response
6. Question Revision

The concept adapted for the scoring of the answer to the question is based on the below figure.

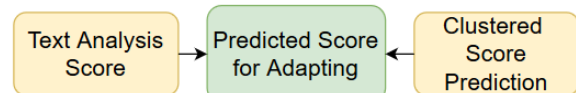


Fig. 14. Scoring Scheme

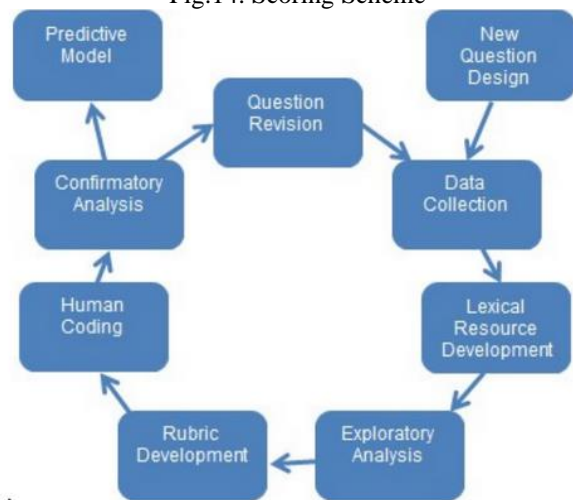


Fig. 15. Evaluation Cycle

K. Lexical Analysis

Lexical analysis is the first half of the grading module, where the responses are lexically analyzed to predict the score that it deserves.

The response from the student is lemmatized to remove all the stop words. This process helps in eliminating the vocabulary that is not needed. Out of the ones that remain, they are sent through a bag-of-words filter where the confidence in the ideas expected is obtained.

The confidence obtained, is a result of a strong rubric that maps the frequency of required words to ideas.

The needed bag of words for each question is stored in the database, which houses the bag-of-words for every question based on its ID

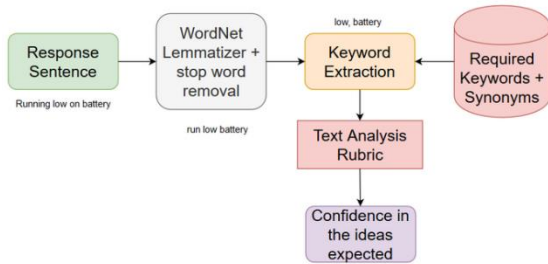


Fig.16. Lexical Analysis

L. Sentence Integrity

Lexical analysis, while gives the confidence for the needed ideas in the response, is significantly flawed. Smart students, who recognize the process can easily respond to questions with nothing more than just keywords. These keywords will be left out by the lemmatizer and thus, will end up in the higher echelons of score classifiers.

To circumvent this issue, putting the response through a Sentence Integrity Checker or Sanity Checker, will help give a better understanding of how well the sentence is structured, and how valid the scores from the Lexical Analysis are.

There are 4 major features that are recognized in this process. They are as follows

1. Avg. noun-to-wordcount ratio
2. Avg. verb-to-wordcount ratio
3. Avg. root count-to-wordcount ratio
4. Actual wordcount-to-expected wordcount

These four parameters give a fair understanding of the structural integrity of the sentence.

On a dataset of over 1000 sentences that were labeled, the observation for the sentence integrity analysis were as follows.

1. ~30% of the words in the sentence should Nouns
2. ~10% of words in the sentence should be Verbs
3. <20% of words in the sentence should be Roots

While these are not hard and fast rules, the outliers are very few and this hypothesis can be worked in as a future enhancement when the data grows.

M. Text Analysis Overview

Marrying the lexical analysis with the Sentence Integrity Checker, robust text analysis can be done on the responses. As the sentence integrity checking is still a primitive hypothesis, it's weightage in the process is not significant. This needs to change as the hypothesis is proved.

The overview of the process is as follows

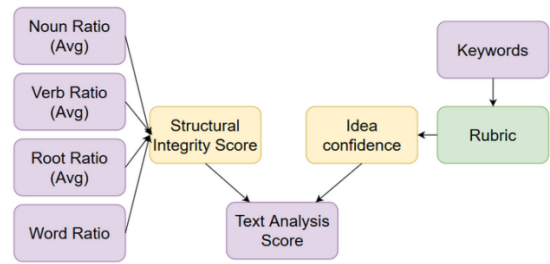


Fig. 17. Text Analysis Overview

N. Artificial Neural Networks and NLP

Artificial Neural Networks, especially Recurring Neural Networks are very effective in processing sequential information. In this scenario, An RNN helps by recursively applying computation to input sequences every instance. The following figure illustrates the same

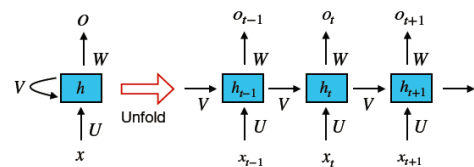


Fig. 18. Illustration of Computation of RNN

Neural networks are well-suited for an application like this, as the potential of data that can be collected is immense, and NLP problems are sequential in nature.

Training this neural network doesn't require a strong feature discovery policy, as keywords are the most important features of a response when vectorized using a bi-gram vectorizer.

Currently, as the data that can be collected is limited due to the offline nature of examinations, getting students to take up examinations online, will solve the data-draught problem that is faced.

After collecting multiple types of answers for each question, building an RNN model with very good accuracies is the first step in the future enhancement of the project.

Until that state is reached, Text analysis will assist teachers in evaluating the responses

V. RESULTS AND DISCUSSION

This section displays the result obtained from the proposed system.

- Home Screen

The starting screen that is displayed initially to the students and faculties.

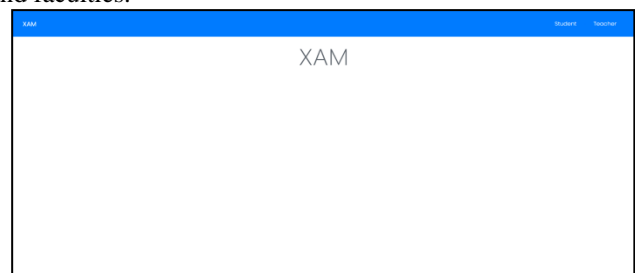
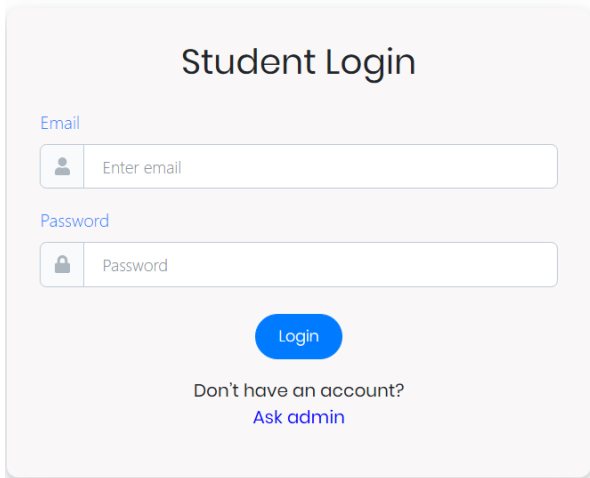


Fig. 19. Home Screen

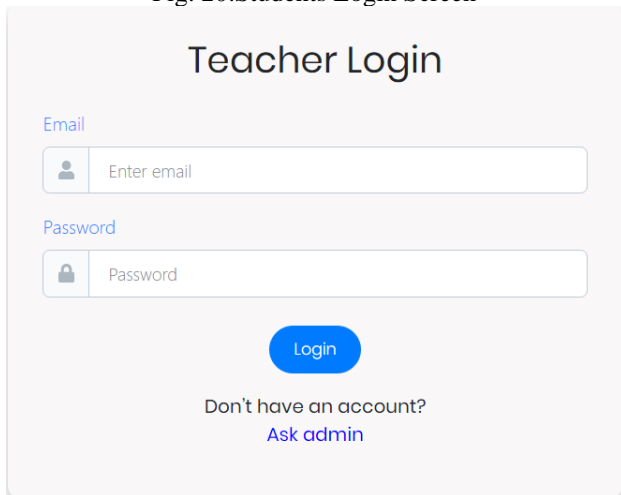
• Login screens

Login credentials are provided to the students and faculties by the admin.



The Student Login screen features a white background with a blue header. It contains two input fields: 'Email' with a person icon and 'Password' with a lock icon. A blue 'Login' button is centered below the fields. At the bottom, there is a link that says 'Don't have an account? Ask admin'.

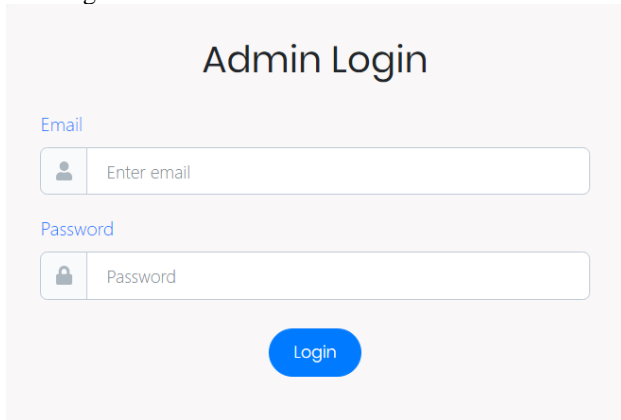
Fig. 20. Students Login Screen



The Teacher Login screen is identical in layout to the Student Login screen, with the same input fields and 'Login' button. The link at the bottom says 'Don't have an account? Ask admin'.

Fig.21. Teachers Login

Admin login is accessed via remote address.

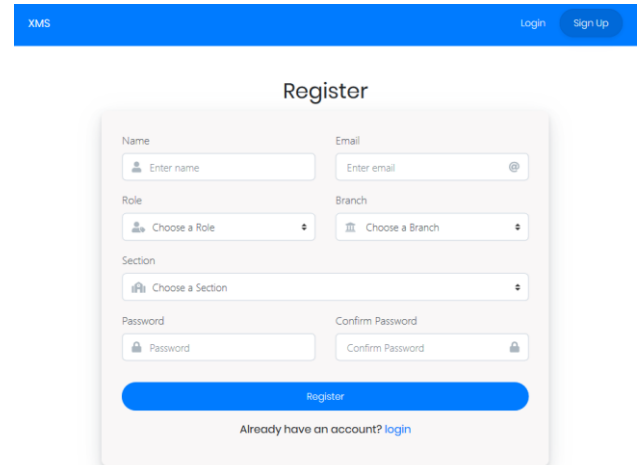


The Admin Login screen is identical in layout to the other login screens, with the same input fields and 'Login' button. The link at the bottom says 'Don't have an account? Ask admin'.

Fig. 22. Admin Login

• Registration Screen

Registering the test takers by the admin, by entering the respective test taker's details.

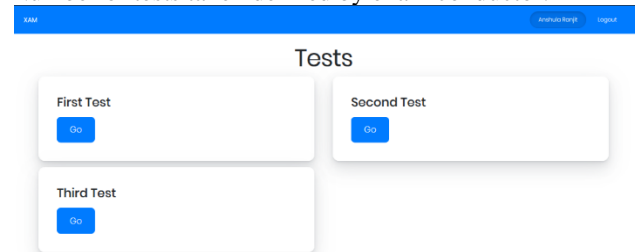


The Register form is located on a blue header with 'XMS' on the left and 'Login Sign Up' on the right. The form itself is white and contains several fields: 'Name' (Enter name), 'Email' (Enter email), 'Role' (Choose a Role), 'Branch' (Choose a Branch), 'Section' (Choose a Section), 'Password' (Password), and 'Confirm Password' (Confirm Password). A blue 'Register' button is at the bottom, with a link 'Already have an account? login' below it.

Fig.23. Register of Test Takers by Admin

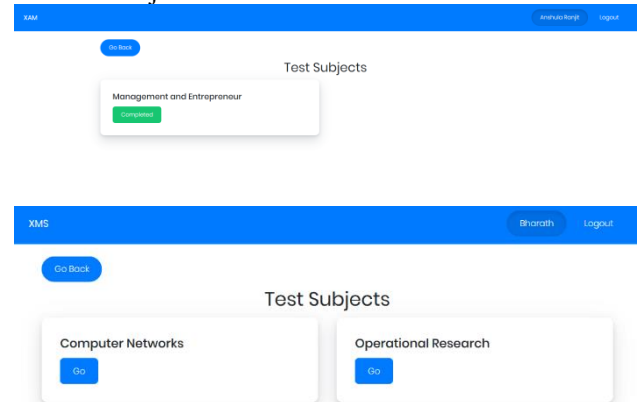
• Assessment Screen

Number of tests taken defined by exam conductor.



The Tests screen has a blue header with 'XMS' and 'Admin Login Logout'. It displays three test cards: 'First Test', 'Second Test', and 'Third Test', each with a blue 'Go' button.

Fig. 24. No. of Test Taken Defined by Exam Conductor
Number of subjects that the test taker should take the test.

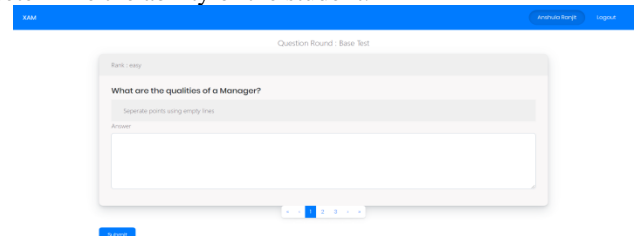


The Test Subjects screen has a blue header with 'XMS' and 'Admin Login Logout'. It shows a 'Go Back' button and a 'Test Subjects' section with two cards: 'Management and Entrepreneur' (Completed) and 'Computer Networks' (Go). Below this is another 'Test Subjects' section with two cards: 'Computer Networks' (Go) and 'Operational Research' (Go).

Fig. 25. No. of Subjects That the Test Taker Should Take the Test

• Examination Screen

The first screen will be the base test question which is used to determine the ability of the student.



The Question Round - Base Test screen has a blue header with 'XMS' and 'Admin Login Logout'. It displays a question: 'What are the qualities of a Manager?' with instructions 'Separate points using empty lines'. Below the question is a text input area and a 'Submit' button.

Fig. 26. Loaded 3 Base Test Questions with Various Difficulty Levels

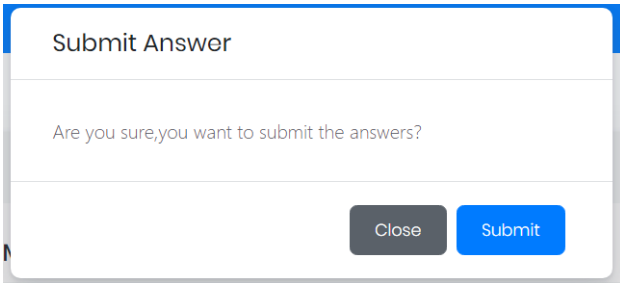


Fig. 27. Submit Modal by Clicking on Submit Answer Button in Test Question Screen

Second round Question is presented by the application, based on grading of the questions answered in base test round.

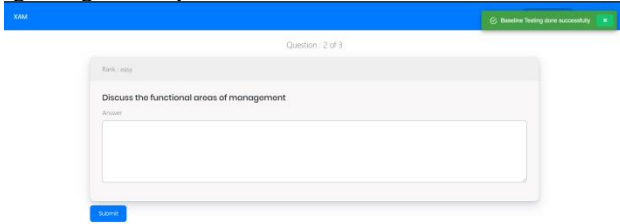


Fig. 28. Next Round of Question by Answering All the Questions from Base Test Questions

The green button Completed indicates the following subject Management and Entrepreneur is completed

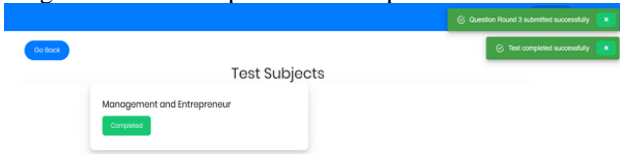


Fig. 29. Indicates the Following Subject Management and Entrepreneur is Completed

• Teacher Evaluation

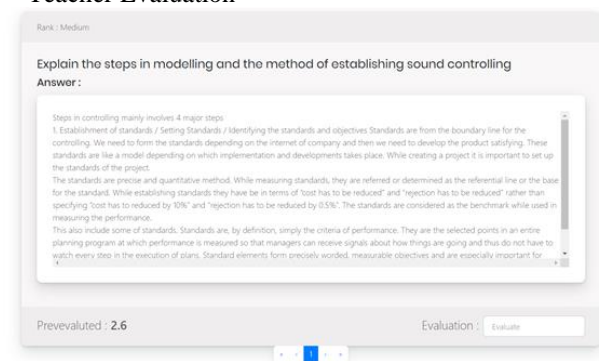


Fig. 30. Indicates the evaluation given by the system

• Student List for the Faculty

Test Students : ME

USN	Name	Prevaluated Results	Evaluation
1	Bharath Karanth A	15	Completed
1	Ashwin Prasad A	15	Rejected
1	Anshula Rarjit	15	Rejected
1	Sarthak Sharma	15	Rejected

Fig. 31.Indicates Student List with Their Pre-evaluated Marks

VI. CONCLUSION AND FUTURE ENHANCEMENT

A smart model for assessing a student is being presented. Instead of using the old archaic ways of measuring the performance of the test taker, we are shifting towards using the current technologies, one should be able to tailor a testing experience pertaining to the traits of the test taker, individually. One can argue that the testing system that is employed in the current society is not an accurate measure of one’s ability or trait. With the growth in technology, it’s only fair that the testing system employed should be made better too. Hence, we call for a better way of testing one’s ability with the use of this system.

One can argue that the testing system that is employed in the current society is not an accurate measure of one’s ability or trait. With the growth in technology, it’s only fair that the testing system employed should be made better too hence, we call for a better way of testing one’s ability with the use of this system. Usage of better machine learning algorithms for the evaluation module will generate a more accurate result, and building a large dataset for the training of the system can result in better application of the machine learning algorithms.

REFERENCES

- [1] Hartman, D. (2019, January 25). Advantages & Disadvantages of Traditional Assessment. Bizfluent. <https://bizfluent.com/info-8475094-advantages-disadvantages-traditional-assessment.html>
- [2] Ozalp-Yaman, S., &Cagiltay, N. E. (2010). Paper-based versus computer-based testing in engineering education IEEE EDUCON 2010 Conference.
- [3] Thakur, D. (n.d.). Principles of Software Design & Concepts in Software Engineering. Ecomputernotes. <https://ecomputernotes.com/software-engineering/principles-of-software-design-and-concepts>
- [4] National Council on Measurement in Education http://www.ncme.org/ncme/NCME/Resource_Center/Glossary/NCME/Resource_Center/Glossary1.aspx?hkey=4bb87415-44dc-4088-9ed9-e8515326a061#anchor1 2017-07-22 at the Wayback Machine
- [5] Kadiyala, V. (2009, February 4). Sequence Diagram [Online forum post]. Dotnetspider.Com. <https://www.dotnetspider.com/forum/191251-Sequence-Diagram>
- [6] Chatterjee, A., & Rani, L. (2013, May–June). Present Evaluation Method of Examination: A Critical Survey. IOSR Journal Of Humanities And Social Science (IOSR-JHSS), 11(1), 46. <https://pdfs.semanticscholar.org/e03d/ffe7f98b95fbf0d26fe054f75c57b794e68c.pdf>
- [7] Wikipedia contributors. (2020, April 22). High-level design. Wikipedia. [https://en.wikipedia.org/wiki/High-level_design#:~:text=High%2Dlevel%20design%20\(HLD\),the%20product%20and%20their%20interfaces.](https://en.wikipedia.org/wiki/High-level_design#:~:text=High%2Dlevel%20design%20(HLD),the%20product%20and%20their%20interfaces.)
- [8] Stephens, R. (2015). Beginning Software Engineering (1st ed.). Wrox.