**PROPOSED TOPIC:**

DEVELOPMENT OF AN INTELLIGENT SOFTWARE AGENT BASED SYSTEM FOR PROCTORING COMPUTER BASED EXAMINATION IN TERTIARY INSTITUTION

**Three active users requirements:**

* Admin
* Examinee or student
* Software Agent
* **Software Requirement Specification**

**ADMIN USER REQUIREMENTS**

* Login using a 2-factor authentication system (Biometric fingerprint + username & password)
* Logout
* Reset users password
* Upload bulk questions
* Upload matching answers
* Register user for examination
* Start and end examination based on set timer
* Start, Pause, Resume and Stop the Software Agent
* Download students examination grade / reports in Excel / Word /PDF formats
* Print student grade / reports in Excel / Word /PDF formats
* Add / Register students for a particular examination
* Search, edit and update students information
* Broadcast short message to all students during examination
* View student result per course taken
* Create default passwords for all students
* Set timer for each course
* Profile students by registration for the software agent usage with fields such as follows:
* First name
* Middle name
* Surname
* Date of Birth
* Town
* LGA
* State of origin
* Mother’s maiden name
* Pet’s name
* Primary school attended
* Secondary school attended
* Favorite food
* Favorite sport
* Phone number
* Active email address, etc, etc

**STUDENT USERS REQUIREMENT**

* Login
* Take Examination
* View result
* Log out

**SOFTWARE AGENT’S REQUIREMENT**

A **software agent**is a computer program that acts for a user or other program in a relationship of agency.

Agents are colloquially known as *bots*, from *robot*. They may be embodied, as when execution is paired with a robot body, or as software such as a chatbot executing on a phone or other computing device. Software agents may be autonomous or work together with other agents or people. Software agents can interact with people (e.g. chatbot, human-robot interaction environments) may also possess human-like qualities such as natural language understanding and speech, personality or embody humanoid form.

The **software agent** should be designed to operate in a **four state life cycle** as it relates with its **environment** which is the **CBT application**.

**[Start] [Pause] [Resume] [Stop]**

The agent should automatically perform the following action in order to detect impersonators during examination condition whenever it is in **Start** state.

* The **software agent** **blocks or freezes** the running **CBT application** which is its environment and superimpose it another question screen, in order to prevent the examinee or students from using the CBT application at that particular point in time, and until the student answers the agent’s question before it will give control to the CBT application to be active again for the students usage.
* The software agent will randomly select questions built from that particular students profile using an **efficient randomized algorithm**. Note: the agent post only one question at a time
* The software agent carefully watches out for the user response to the question asked. It takes cognizance of the response time and the correctness of the answer given.
* The agent should be able to reason correctly per question asked in order to correctly classify or cluster the examinee or student into **“Normal”, “Moderate Treat” or “High Treat or Risk”** based on the usage of supervised or unsupervised machine learning algorithm. Note the agent sets a particular threshold value for the various classification or clusters.
* The agent should ask the examinee questions at certain interval before concluding or given its final **VERDICT**
* The agent should automatically logout the examinee or student from the CBT application immediately it **detects** the **risk factor is high**
* The agent should send or commit all entries or activities noticed to a **Log file**
* The agent should be able to send quick **feedback message** to the administrator in-charge about the **suspected case of impersonation**.
* etc

**SOFTWARE REQUIREMENT SPECIFICATION**

**FUNCTIONAL**

* The system should support cryptographic technique like MD5, SHA2, DES, 2DES, 3DES, AES, or hybrid technique by combining any two or three techniques etc to guarantee security of the data on the system. In other words, the examination questions must remain encrypted in the system until whenever any authenticated examinee or student is granted access to the platform during examination
* The system should use an efficient **randomized algorithm** to handle random administration of Multiple Choice Questions (MCQ) to reduce the level of examination malpractice.
* The system should use an **efficient grading algorithm** capable of matching answered questions to the correct answers, and also capable of computing the overall score correctly in order to eliminate or reduce false alarm rates like false positive and false negative. (Confusion Matrix).
* The system should use **effective Algorithm for Mass Auto-Submission** of examination on expiration of set time.
* The system should be able to support auto-grading of students and examination results report generation
* The system should be able to integrate students continuous assessment (CA) module by computing the CA to the Examination score to give the overall score
* The system should tick the answer button [/ ] immediately each question is answered.
* The system should support a two factor authentication module for admin user i.e. (Biometric fingerprint + Secure Username & Password system), only username and password for ordinary users.

**NON-FUNCTIONAL**

* The system should support the integration of the possibility of supporting both multiple questions and single questions generation in a single screen view or display.
* The system should be able to display the total number of questions to be answered on either the left hand side of the screen display in a matrix format made up of rows and columns or the bottom bar made up of only rows
* The system should be able to support BACKUP/RESTORATION of files in case of unforeseen circumstances
* The system should be users friendly in terms of navigational bars, scroll bars, etc with attractive GUI and Users Experience (UX) environment
* The system should support scalability on mobile devices