International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 3 | ISSN : 2456-3307

Construct an Android based Appliance for Sheltered Assessment Management Scheme (SAMS)

J. Jelsteen[1], K.Arjun[2], P.P. Athul Babu[2]
[1]Assistant Professor, Nehru College of Management,Coimbatore, Tamil Nadu, India
[2]MCA Student, Nehru College of Management, Coimbatore, Tamil Nadu, India

ABSTRACT

M-Learning has better the e-learning by creation the wisdom method student centered. However, enforcing exam protection in open environments where each student has his/her own mobile/tablet device connected to a Wi-Fi network through which it is further connected to the Internet can be one of the most tough tasks. In such environments, students can easily swap information over the network during exam time. This paper aims to classify different vulnerabilities that may abuse exam security in m-learning environments and to propose the appropriate security services and countermeasures that can be put in place to ensure exam security. It also aims to amalgamate the resulting secure exam structure with an existing, open source and widely established Learning Management System (LMS) and its service extension to the m-learning environment, namely "the Moodbile Project".

Keywords: Sheltered, Assessment , Management , Scheme

I. INTRODUCTION

In this project the creation of an android application that allows to attend the exam securely. It uses the mobile cameras for capturing student image and QR scanner to scanning the QR provided in the exam hall. The students login to the android application with their username and password, and then QR is scanned, to authenticate students the image is captured and check for match. If the student is authenticated successfully, he/she can attend the exam. The result is an android application that can authenticate students and allow attending the exam.

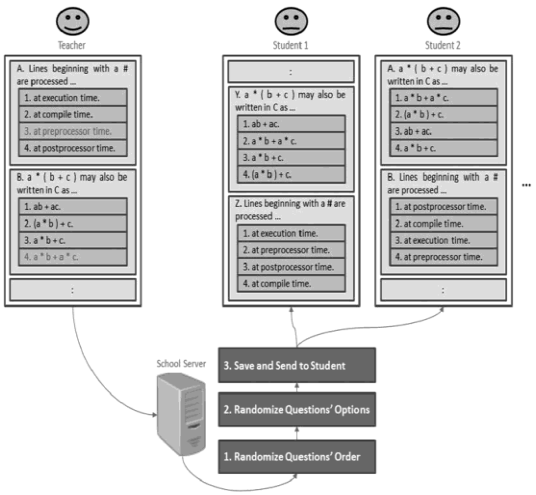

Figure 1. Architecture

II. IMPLEMENTATION

Implementation is the process of converting a system into an operational one. The designed system is converted to an operational one using a suitable programming language Installation of the system takes place only after it is found to be error free.

Implementation walkthroughs ensure that the completed system actually accomplishes the original requirement. This walkthrough occurs just before the system goes into use, and it includes careful review of all manuals, training materials and system documentation.

In Student seat allocation, the exam server hashes the access string(contain hall number, seat number, subject id and student id) using SHA-1 to produce the message digest. Then this message digest is signed with RSA private key. Finally, the exam server generates the QR-Code of the signed access string to get the exam access token.

Find the weight vector of the stored images: To initialize the recognition task, at first the weight vector of the training images should be derived and calculated. And these vectors have to be stored corresponding to respective training image. Load testing image: Then the testing image will be loaded by the user and the system will convert this into a vector. Before do this some pre-processing operations should be handled and done to the testing image in order to improve the quality of the image to support to do better recognition of face. Find feature vector: After loaded the training image, the feature vector of the training image can be extracted and found. Here the new facial image is transformed into its eigen face components by a well-defined simple operation.

III. USER MANUAL PROCUDURE

At the beginning of the development phase, the initial implementation plan is created to schedule and manage the many different activities that must be integrated to the plan. The implementation is updated throughout the development phase, integrating in a changeover plan for the operation phase. Evaluation of the system is performed to identify its strengths and weakness. The actual evaluation can occur along any of the following dimensions.

Operational Evaluation: assessment of the manner in which the system functions, including ease of use, response time, suitability of information formats, overall reliability, and level of utilization.

Organization Impact: Identification and measurement of benefits to the organization in such areas as financial concerns operational efficiency, and competitive impact. Includes impact on internal and external information flows.

User Manager Assessment: Evaluation of the attitudes of senior and user mangers within the organization, as well as end-users.

Development Performance: Evaluation of the development process in accordance with such yardsticks as overall development time and effort

IV. TESTING METHODOLIGIES

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

V. STRATEGIC APPROACH TO SOFTWARE TESTING

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis

where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

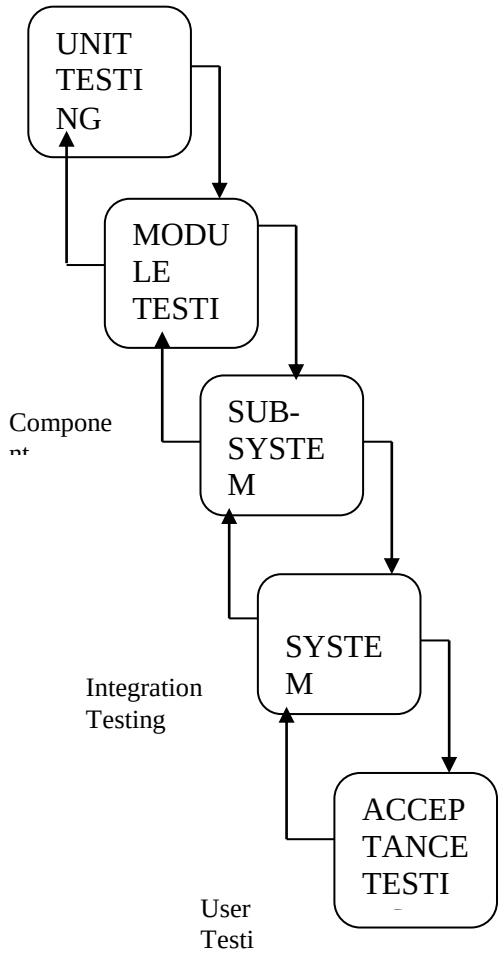

Figure 2. Testing Methodologies

VI. SOFTWARE MAINTANENCE

The term 'software maintenance' is used to describe the software engineering activities that occur delivery of a software product to the customer. The maintenance phase of the software life cycle is the time period in which a software product performs useful work. Maintenance activities involve making enhancements to software products, adapting products to new environments and correcting problems are defined by describing four activities that are undertaken after a program is released for use. The four different maintenance activities: corrective maintenance, adaptive maintenance, perfect maintenance or enhancement and preventive maintenance or reengineering. Only about 20per cent of all maintenance work is spent adapting existing systems to changes in their external environment, making enhancements requested by the users, and reengineering an application for future use. When maintenance is considered to encompass all these activities, it is relatively easy to see why it absorbs too much effort.

Maintenance involves the software industry captive, typing up system resources. It means restoring something to its original condition. Maintenance involved a wide range of activities including correcting, coding and design errors, updating documentation and test data and upgrading user support. Maintenance was done after the successful implementation. Maintenance is continued till the product is re-engineered or deployed to another platform. Maintenance is also done based on fixing the problems reported, changing the interface with other software or hardware enhancing the software.

The developed system is supposed to used in an engineering consultancy organization for coordinating various activities of the project. An organization that is making a commitment to a software system must be assured that the software system will continue to be available and that will be maintained to suit the changing requirements. System maintenance can be termed as an enigma of system development. It holds the software industry captive, toping up programming

ELEMENTS:

Student Elements
Teacher Elements
Admin Elements

VII. ELEMENTSS DESCSRIPTION

Student Elements:

- ✓ This is developed in Client Side (Android).

- ✓ Here students register and logins to write exams.
- ✓ Students by typing exam name can write the appropriate exams.
- ✓ Students after writing exams can view their results using the Secret key generated at the time of registration..
- ✓ Student have to memorise their secret key to view their own results

Teacher Elements:

- ✓ This Element is developed in Client Side (Android).
- ✓ View all the student results.
- ✓ Schedules exam and inform the students about exam by sending messages through WHATSAPP

Admin Elements:

- ✓ This element is developed in Server Side.
- ✓ View all the student details.
- ✓ Immediately removes the student name from table after the completion of exams.
- ✓ So the student can't access again.
- ✓ Views all the student results and can remove the secKey so that student can view.
- ✓ The result only for particular period of time.
- ✓ Views all the teachers details

VIII. RESULTS

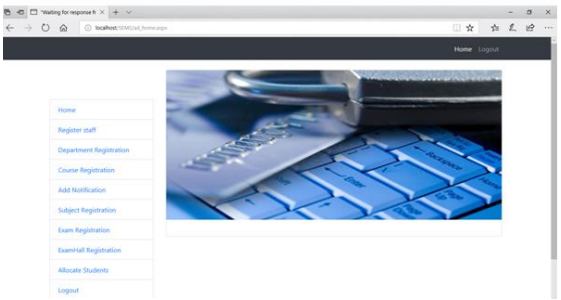

Figure 3: Architecture

IX. FUTURE ENHANCEMENT

Secure Management System provide authentication of students using any of the well-known secure authentication mechanisms. Students might still attempt to cheat by simply exchanging their mobile/tablet devices after they get authenticated by the Exam Server. Top this issue, the system triestore-authenticate the student biometrically by asking the present their faces in front of the mobile camera on a random basis

X. CONCLUSION

Today smart phones running Android Operating System are popular. They are cheap but with considerably good performance. Secure Management System provide authentication of students using any of the well-known secure authentication mechanisms. The proposed system provide a Secure Exam Management System to mitigate the unique exam security threats that existing M-learning environment.

XI. REFERENCES

[1]. E.R.Weippl," Security considerationsin M-learning: Threatsand countermeasures,"Adv. Technol.Learn., vol. 4, no. 2, pp. 99-105, Jan. 2007.

[2]. Z.Ugray," Security and privacy issues in mobile learning,"Int.J.MobileLearn.Org.,vol.3, no. 2, pp. 202-218,Apr.2009.

[3]. G. Frankl, P. Schartner, and G. Zebedin," Secure online examsusingstudents'devices,"in Proc. IEEE GlobalEng. Edu.Conf.,Apr.2012.