

# **SMART PHONE BASED REMOTE MONITORING TOOL FOR E-LEARNING**

Submitted in partial fulfillment of the requirements for the award of  
Bachelor of Science degree in Computer Science

By

**MOHAN.S (39290068)**

**PREM.K (39290081)**



**DEPARTMENT OF COMPUTER SCIENCE**

**SCHOOL OF COMPUTING**

## **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade “A” by NAAC**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,  
CHENNAI - 600 119  
MARCH - 2022**



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)



## DEPARTMENT OF COMPUTER SCIENCE

### BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **MOHAN.S (39290068)** and **PREM.K (39290081)** who have done the Project work as a team who carried out the project entitled "**SMART PHONE BASED REMOTE MONITORING TOOL E-LEARNING**" under my supervision from November 2021 to March 2022.

#### Internal Guide

**Dr. M. MAHESHWARI M.E., PHD**

**Head of the Department**

Submitted for Viva voice Examination held on \_\_\_\_\_

Internal Examiner

External Examiner

## **DECLARATION**

I **MOHAN.S** hereby declare that the Project Report entitled **SMART PHONEBASED REMOTE MONITORING TOOL FOR E-LERNING** done by me under the guidance of **DR.M.MAHESHWARI M.E., PHD** is submitted in partial fulfillment of the requirements for the award of Bachelor of Science degree in computer science .

**DATE:**

**PLACE: CHENNAI**

**SIGNATURE OF THE CANDIDATE**

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully.

I am grateful to them. I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.,** and **Dr. S. Vigneshwari M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. M. Maheshwari M.E., PHD** for his valuable guidance suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

## **ABSTRACT**

In this paper, a smartphone-based learning monitoring system is presented. During pandemics, most of the parents are not used to simultaneously deal with their home of\_ice activities and the monitoring of the home school activities of their children.

Therefore, a system allowing a parent, teacher or tutor to assign task and its corresponding execution time to children, could be helpful in this situation. In this work, a mobile application to assign academic tasks to a child, measure execution time, and monitor the child's attention, is proposed. The children are the users of a mobile application, hosted on a smartphone or tablet device, that displays an assigned task and keeps track of the time consumed by the child to perform this task.

Time measurement is performed using face recognition, so it is possible to infer the attention of the child based on the presence or absence of a face. The app also measures the time that the application was in the foreground, as well as the time that the application was sent to the background, to measure boredom.

The parent or teacher assigns a task using a desktop application speci\_cally designed for this purpose. At the end of the time set by the user, the application sends to the parent or teacher statistics about the execution time of the task and the degree of attention of the child.

## TABLE OF CONTENTS

| CHAPTER<br>No | TITLE                               | PAGE<br>No |
|---------------|-------------------------------------|------------|
|               | <b>LIST OF FIGURES</b>              | <b>1</b>   |
|               | <b>LIST OF ABBREVIATIONS</b>        | <b>2</b>   |
| <b>1</b>      | <b>INTRODUCTION</b>                 |            |
| 1.1           | About the Project                   | <b>3</b>   |
| <b>2</b>      | <b>SYSTEM ANALYSIS</b>              |            |
| 2.1           | Existing system                     | <b>4</b>   |
| 2.2           | Proposed system                     | <b>5</b>   |
| <b>3</b>      | <b>REQUIREMENTS SPECIFICATION</b>   |            |
| 3.1           | Introduction                        | <b>6</b>   |
| 3.2           | Hardware and Software specification | <b>8</b>   |
| 3.3           | Technologies Used                   | <b>9</b>   |
| 3.3.1         | Java                                | <b>9</b>   |
| 3.3.3.1       | Introduction to Java                | <b>10</b>  |
| 3.3.1.2       | Working of Java                     | <b>11</b>  |
| <b>4</b>      | <b>Project Purpose and Scope</b>    |            |
| 4.1           | Purpose                             | <b>22</b>  |
| 4.2           | Project Scope                       | <b>22</b>  |

|          |                                       |           |
|----------|---------------------------------------|-----------|
| 4.3      | Product Perspective                   | 22        |
| 4.4      | System Features                       | 22        |
| 4.5      | Design and Implementation Constraints | 23        |
| 4.6      | Other Nonfunctional Requirements      | 23        |
| <b>5</b> | <b>SYSTEM DESIGN</b>                  |           |
| 5.1      | Architecture Diagram                  | 25        |
| <b>6</b> | <b>SYSTEM DESIGN – DETAILED</b>       |           |
| 6.1      | Modules                               | 34        |
| 6.2      | Module explanation                    | 35        |
| <b>7</b> | <b>CODING AND TESTING</b>             |           |
| 7.1      | Coding                                | 36        |
| 7.2      | Coding standards                      | 36        |
| 7.3      | Test procedure                        | 38        |
| 7.4      | Test data and output                  | 38        |
|          | <b>SOURCE CODE</b>                    | <b>45</b> |
|          | <b>SNAP SHOTS</b>                     | <b>74</b> |
|          | <b>REFERENCE</b>                      | <b>86</b> |

## **LIST OF FIGURES**

5.1 System Design

5.2 Sequence Diagram

5.3 Use Case Diagram

5.4 Activity Diagram

5.5 Collaboration Diagram

5.6 Data Flow Diagram

5.7 Class Diagram

## LIST OF ABBREVIATIONS

|             |                               |
|-------------|-------------------------------|
| <b>JDK</b>  | Java Development Toolkit      |
| <b>DEX</b>  | Dalvik Executables            |
| <b>TCP</b>  | Transmission Control Protocol |
| <b>IP</b>   | Internet Protocol             |
| <b>HTTP</b> | Hyper Text Transfer Protocol  |
| <b>ADT</b>  | Android Development Tool      |

# CHAPTER 1

## INTRODUCTION

### **Aim:**

To monitor the student's attention on their school activities and to make the remote learning methods more effective.

### **Synopsis:**

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. To analyze the degree of attention of the students of each instance, it needs to be trained using machine learning algorithms. By using this algorithm, we can monitor the activities of the students and provides results to their parents based on the degree of attention of each and every student.

## **CHAPTER 2**

### **SYSTEM ANALYSIS**

#### **2.1 EXISTING SYSTEM**

In the existing system, it is difficult to monitor the students during their online exams/daily tests and online classes by their parents, teacher(s) or tutor(s). This may cause a pandemic situation for each and every parent or tutor to know about their children and to grab their attention towards studies. During online classes, the staff may teach the subjects and give some activities through the online class based on the students grade. Some students may complete but some may not. The parents who are working from home may busy with their office activities and may not concentrate on their children. In the existing system, we are not having any methods or ideas to make students to complete these activities within the given time with maximum attention.

#### ***Problem Definition***

Many tasks are assigned to the children who are attending their online classes but there is no idea about their completion of their activities given by their staffs. Most of the parents are busy with their home and office activities they may not monitor their children who are attending the online classes or preparing for their exams.

## **2.2PROPOSED SYSTEM**

In this system, we propose a remote learning system and make it more efficient and more useful for parents, students and staffs. The staff will register their details. The Admin have to give permission for the registered staff to proceed with their login process. If the admin reject a staff, then that staff will not be accessible with their further process. The authorised staff will proceed their process after logging in. They will add the students along with their personal details who are under his/her class.

Then the staff will upload tasks or activities under courses with any number of selected students who need to attend those activities. The students will register with their android mobile phone. Once the course get started, mail will be sent through their parent's email. After getting mail, students will login using their registered email and the password which gets generated while staffs adding students and it will be sent through the parent's email and the student will start the activity. After completing every activity, the image-based evidence and the video of student while doing activity(attention video) will be saved. The staffs can view the created courses and activities. The students who are added in the activity will get displayed inside that activity. For every activity, the staff can view the performance of the student i.e., the image-based evidence after their activity completion. The attention will get calculated using the attention video in python. After viewing the evidence and the percentage of attention, the staff will enter the mark based on their evidence and attention. Finally, the result will be sent to the parents.

### **Advantages**

1. This system makes the remote learning methods more effective and helpful for staffs, students and parents.
2. In this method, the students will perform in their exams, activities and in their online classes with more attention than before.

## CHAPTER 3

### REQUIREMENT SPECIFICATIONS

#### 3.1 INTRODUCTION

In this paper, a tool to monitor the learning activities of children whose parents are working in the home is proposed. The developed tool monitors the attention levels of children solving assigned tasks that cannot be supervised by a present adult. The monitoring information is useful for both, parents and teachers, who can use it to make decisions about the effectiveness of remote learning methods. The proposed tools are focused on handwritten tasks, such as solving arithmetic or algebraic operations, writing some or drawing, where children do not directly interact with the smartphone/tablet, and use it only to read the task description and to report image-based evidence of the carried out work, which will be later revised by a teacher or a parent. Several approaches have been introduced in order to develop software tools for different gaze estimation purposes in e-learning. In 2018 Steilet *et al.* [1], presented a work related to the task of predicting users' gaze behaviour (overt visual attention) in the near future. Kangas *et al.* [2] described a study of combining gaze gestures with vibrotactile feedback. In this study, gaze gestures were used as input for a mobile device and vibrotactile feedback as a new alternative way to give confirmation of interaction events. Results show that vibrotactile feedback significantly improved the use of gaze. Tonsenet *et al.* [3] proposed an eye tracking tool that uses millimeter-size RGB cameras that can be fully embedded into normal glasses. To compensate for the cameras' low image resolution of only a few pixels, the proposed approach uses multiple cameras to capture different views of the eye, as well as learning-based gaze estimation to regress from eye images to gaze directions directly. Mesquita and Lopes [4] proposed an image processing system to provide a valuable aid to kindergarten teachers, helping them in the registration of observation, by automatically detecting and measuring head posture, including time records. In 2014 Teyebet *et al.* [5] presented a

drowsiness detection based on video approach to analyze two visual driver's signs: eyes blinking and head movement.

Hutanu and Berteau in 2019 [6], described that one of the most accurate methods to record user behavior is through eye tracking technology. Eye tracking is a method that records real, natural and objective user behavior. Eye movements are and subconscious, may reveal information that is not accessible even to the responder. The human gaze can tell exactly what has been seen, in what order and for how long and, on the other side, what has been missed. Therefore, eye tracking data valuable insights for improving the learning process. Chonget *et al.* in 2018 [7] proposed automated methods for and quantifying visual attention from images and video of gaze following. Gaze following can be applied to identify, categorize quick glances to objects, and even identify when someone is not paying attention. Copeland [8] stated that using eye gaze to control adaptive learning environments provides the ability to go beyond the student's surface answering behavior or preferences and adapt to the student's implicit behavior. Thus tracking has been shown to be a powerful tool for investigating how humans interact with computer interfaces. In 2018 Waleed *et al.* [9], introduced an accurate, reliable and a low-cost system based on laptop webcam for eye-gaze estimation. Ricciardelli and Driver [10] described the impact of different sources of information by examining the role of head orientation in gaze-direction judgments, based on past studies, where head orientation may be combined with perceived gaze direction in judging where someone else is attending. Fathiet *et al.* [11] presented a probabilistic generative model for simultaneously recognizing daily actions and predicting gaze locations in videos recorded from an egocentric camera. In 2016 Jiang *et al.* [12] proposed a system to early detect the visual attention of users. Acquiring the location of the intended object is made by gazing at the intended object and holding up the smartphone so that the object as well as user's face can be simultaneously captured by the front and rear cameras. Xiao and Feng [13] introduced a driver attention detection system based on smartphones with dual cameras. In this work, a feature detector describing pupil location, yaw and pitch angle of eyes, position, and size of the face detected with the front camera is proposed. This, permits to track drivers gaze

direction, to detect nine safe gaze areas and one unsafe gaze areas. Simultaneously, the view of road is detected with the rear camera using optical flow methodology combined dynamic background compensation.

In 2019 Jiang *et al.* [14], introduced efficient algorithms to obtain both the distance between the camera and user, the user's gaze direction and the object's direction from the. Xu and Teng [15] states that the orientation of the reflects the direction of attention, but changes in posture space can interfere with the direction of attention. With the goal of solving this problem, a scoring module that takes the head Euler angle in three directions as input, and introduces spatial information to correct the angle is proposed. Asteriadiset *al.* [16] explored the role of head rotation and eye gaze directionality to the human perception of attention from non-verbal cues from a theoretical and technical of view. Sana *et al.* [17] examined the effects of in-class laptop use on student learning in a simulated classroom. The authors concluded that participants who multitasked on a laptop during a lecture scored lower on a test compared to those who did not multitask, and participants who were in direct view of a multitasking peer scored lower on a test compared to who were not. Recently, in 2020 Støle *et al.* [18] states that children's screen reading comprehension has been underexplored, therefore an experiment for measuring the effect of the reading medium on younger (10-year old) readers' carried out. Chernet *al.* [19] proposed a smartphone based hearing assistive system to facilitate speech recognition for various target users, who could benefit from enhanced listening the clarity in the classroom.

## **3.2 HARDWARE AND SOFTWARE SPECIFICATION**

### **3.2.1 HARDWARE REQUIREMENTS**

- Hard Disk : 80GB and Above
- RAM : 4GB and Above
- Processor : P IV and Above

### **3.2.2 SOFTWARE REQUIREMENTS**

- Window7 and above (64bit)
- JDK 1.8
- Python 3.6.3
- Tomcat 8.5
- MySQL
- Android Studio 3.4
- Android Phone

### **3.3 TECHNOLOGIES USED**

- Programming Language : JAVA
- Core Framework : Collection
- J2EE : Spring Framework
- Script : JavaScript
- Web Designing : HTML5, CSS3
- Animation Effect : JQuery
- Front-End Framework : Bootstrap3

#### **3.3.1 JAVA**

Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. The language, initially called Oak (named after the oak trees outside Gosling's office), was intended to replace C++, although the feature set better resembles that of Objective C.

##### **3.3.1.1 INTRODUCTION TO JAVA**

Java has been around since 1991, developed by a small team of Sun Microsystems developers in a project originally called the Green project. The intent of

the project was to develop a platform-independent software technology that would be used in the consumer electronics industry. The language that the team created was originally called Oak.

The first implementation of Oak was in a PDA-type device called Star Seven (\*7) that consisted of the Oak language, an operating system called GreenOS, a user interface, and hardware. The name \*7 was derived from the telephone sequence that was used in the team's office and that was dialed in order to answer any ringing telephone from any other phone in the office.

Around the time the First Person project was floundering in consumer electronics, a new craze was gaining momentum in America; the craze was called "Web surfing." The World WideWeb, a name applied to the Internet's millions of linked HTML documents was suddenly becoming popular for use by the masses .The reason for this was the introduction of a graphical Web browser called Mosaic, developed by ncSA. The browser simplified Web browsing by combining text and graphics into a single interface to eliminate the need for users to learn many confusing UNIX and DOS commands. Navigating around the Web was much easier using Mosaic.

It has only been since 1994 that Oak technology has been applied to the Web. In 1994, two Sun developers created the first version of Hot Java, and then called Web Runner, which is a graphical browser for the Web that exists today. The browser was coded entirely in the Oak language, by this time called Java. Soon after, the Java compiler was rewritten in the Java language from its original C code, thus proving that Java could be used effectively as an application language. Sun introduced Java in May 1995 at the Sun World 95 convention.

Web surfing has become an enormously popular practice among millions of computer users. Until Java, however, the content of information on the Internet has been a bland series of HTML documents. Web users are hungry for applications that

are interactive, that users can execute no matter what hardware or software platform they are using, and that travel across heterogeneous networks and do not spread viruses to their computers. Java can create such applications.

#### *3.3.1.2 WORKING OF JAVA*

For those who are new to object-oriented programming, the concept of a class will be new to you. Simplistically, a class is the definition for a segment of code that can contain both data (called attributes) and functions (called methods).

When the interpreter executes a class, it looks for a particular method by the name of **main**, which will sound familiar to C programmers. The main method is passed as a parameter an array of strings (similar to the `argv []` of C), and is declared as a static method.

To output text from the program, we execute the **println** method of **System.out**, which is java's output stream. UNIX users will appreciate the theory behind such a stream, as it is actually standard output. For those who are instead used to the Windows platform, it will write the string passed to it to the user's program.

Java consists of two things:

- Programming Language
- Platform

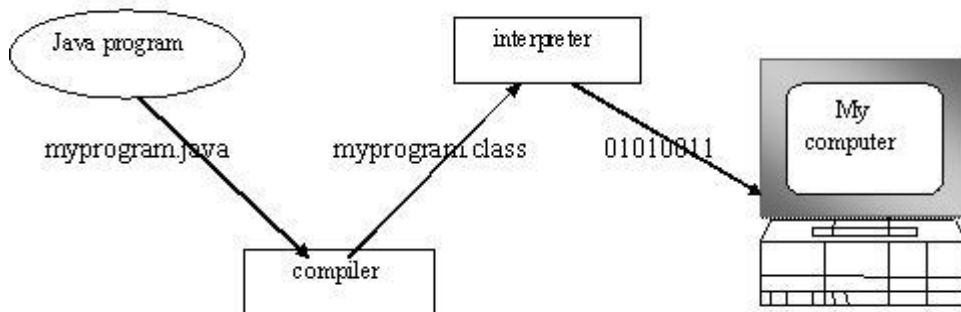
#### *3.3.1.3 THE JAVA PROGRAMMING LANGUAGE*

Java high-level programming language that is all of the following is a

- Simple
- Object Oriented
- Distributed
- Interpreted
- Robust
- Secure
- Architecture-neutral
- Portable
- High-Performance
- Multithreaded
- Dynamic

The code can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

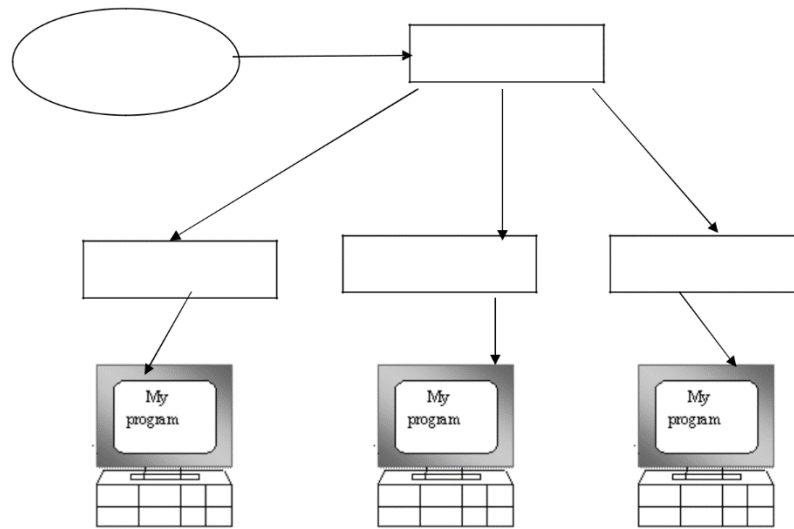
Java is unusual in that each Java program is both compiled and interpreted. With a compiler, you translate a Java program into an intermediate language called **Java byte codes** – the platform independent codes interpreted by the Java interpreter. With an interpreter, each Java byte code instruction is parsed and run on the computer. Compilation happens just once; interpretation occurs each time the program is executed. This figure illustrates how it works:



**Fig.3.1**

You can think of Java byte codes as the machine code instructions for the **Java Virtual Machine (JVM)**. Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of JVM. That JVM can also be implemented in hardware. Java byte codes help make “write once, run anywhere” possible.

You can compile your Java program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the JVM. For example, that same Java program can be run on Windows NT, Solaris and Macintos.



**Fig.3.2**

#### 3.3.1.4 THE JAVA PLATFORM

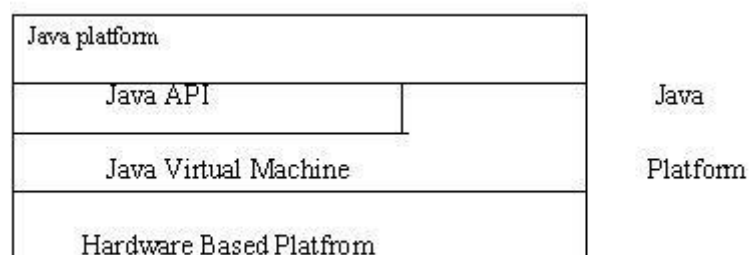
A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other, hardware-based platforms. Most other platforms are described as a combination of hardware and operating system.

The Java platform has two components :

- The Java Virtual Machine (JVM)
- The Application Programming Interface (JAVA API)

You've already been introduced to the JVM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries (**packages**) of related components. The following figure depicts a Java program, such as an application or applet, that's running on the Java platform. As the figure shows, the Java API and Virtual Machine insulates the Java program from hardware dependencies.



**Fig.3.3**

As a platform-independent environment, Java can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte compilers can bring Java's performance close to that of native code without threatening portability.

### **3.3.2 APACHE TOMCAT SERVER**

Apache Tomcat (formerly under the Apache Jakarta Project; Tomcat is now a top level project) is a web container developed at the Apache Software Foundation. Tomcat implements the servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted. Because Tomcat includes its own HTTP server internally, it is also considered a standalone web server.

#### **Environment**

Tomcat is a web server that supports servlets and JSPs. Tomcat comes with the Jasper compiler that compiles JSPs into servlets.

The Tomcat servlet engine is often used in combination with an Apache web server or other web servers. Tomcat can also function as an independent web server. Earlier in its development, the perception existed that standalone Tomcat was only suitable for development environments and other environments with minimal requirements for speed and transaction handling. However, that perception no longer exists; Tomcat is increasingly used as a standalone web server in high-traffic, high-availability environments.

Since its developers wrote Tomcat in Java, it runs on any operating system that has a JVM.

#### **Product features**

Tomcat 3.x (initial release)

- Implement the Servlet 2.2 and JSP 1.1 specifications
- Servlet reloading
- Basic HTTP functionality Tomcat 4.x
- Implement the Servlet 2.3 and JSP 1.2 specifications

- Servlet container redesigned as Catalina
- JSP engine redesigned as Jasper
- Coyote connector
- Java Management Extension (JMX), JSP and Struts-based administration
- Tomcat 5.x
- Implements the servlet 2.4 and JSP 2.0 specification
- Reduced garbage collection, improved performance and scalability
- Native Windows and Unix wrappers for platform integration
- Faster JSP parsing

## **HISTORY**

Tomcat started off as a servlet specification implementation by James Duncan Davidson, a software architect at Sun. He later helped make the project open source and played a key role in its donation by Sun to the Apache Software Foundation.

Davidson had initially hoped that the project would become open-sourced and, since most open-source projects had O'Reilly books associated with them featuring an animal on the cover, he wanted to name the project after an animal. He came up with Tomcat since he reasoned the animal represented something that could take care of and fend for itself. His wish to see an animal cover eventually came true when O'Reilly published their Tomcat book with a tomcat on the cover.

### ***3.3.3 Android Introduction:***

Android is a Linux based operating system it is designed primarily for touch screen mobile devices such as smart phones and tablet computers. The operating system has developed a lot in last 15 years starting from black and white phones to recent smart phones or mini computers. One of the most widely used mobile OS these days is android. The android is software that was founded in Palo Alto of California in 2003.

The android is a powerful operating system and it supports large number of applications in Smartphones. These applications are more comfortable and

advanced for the users. The hardware that supports android software is based on ARM architecture platform. The android is an open source operating system means that it's free and any one can use it. The android has got millions of apps available that can help you managing your life one or other way and it is available low cost in market at that reasons android is very popular.

The android development supports with the full java programming language. Even other packages that are API and JSE are not supported. The first version 1.0 of android development kit (SDK) was released in 2008 and latest updated version is jelly bean.

The android is a operating system and is a stack of software components which is divided into five sections and four main layers that is



**Fig.3.4**

### 3.3.3.1 Linux kernel:

The android uses the powerful Linux kernel and it supports wide range of hardware drivers. The kernel is the heart of the operating system that manages input and output requests from software. This provides basic system functionalities like process management, memory management, device management like camera,

keypad, display etc the kernel handles all the things. The Linux is really good at networking and it is not necessary to interface it to the peripheral hardware. The kernel itself does not interact directly with the user but rather interacts with the shell and other programs as well as with the hardware devices on the system.

### *3.3.3.2 Libraries:*

On top of a Linux kernel there is a set of libraries including open source web browser such as webkit, library libc. These libraries are used to play and record audio and video. The SQLite is a data base which is useful for storage and sharing of application data. The SSL libraries are responsible for internet security etc.

### *3.3.3.3 Android Runtime:*

The android runtime provides a key component called Dalvik Virtual Machine which is a kind of java virtual machine. It is specially designed and optimized for android. The Dalvik VM is the process virtual machine in the android operating system. It is software that runs apps on android devices.

The Dalvik VM makes use of Linux core features like memory management and multithreading which is in a java language. The Dalvik VM enables every android application to run its own process. The Dalvik VM executes the files in the .dex format.

### *3.3.3.4 Application frame work:*

The application framework layer provides many higher level services to applications such as windows manager, view system, package manager, resource manager etc. The application developers are allowed to make use of these services in their application.

### *3.3.3.5 Applications:*

You will find all the android applications at the top layer and you will write your application and install on this layer. Examples of such applications are contacts, books, browsers, services etc. Each application performs a different role in the overall applications.

Advantages:

- Android is Linux based open source operating system, it can be developed by any one.
- Easy access to the android apps
- You can replace the battery and mass storage, disk and UDB option
- It supports all Google services
- The operating system is able to inform you of a new SMS and Emails or latest updates
- It supports Multitasking
- Android phone can also function as a router to share internet
- It is free to customize
- Can install a modified ROM
- It supports 2D and 3D graphics

## **Base 64Introduction:**

Base64 is an encoding algorithm that allows you to transform any characters into an alphabet which consists of Latin letters, digits, plus, and slash. Thanks to it, you can convert Chinese characters, emoji, and even images into a “readable” string, which can be saved or transferred anywhere.

To figuratively understand why Base64 was invented, imagine that during a phone call Alice wants to send an image to Bob. The first problem is that she cannot simply describe how the image looks, because Bob needs an exact copy. In this case, Alice may convert the image into the binary system and dictate to Bob the binary digits (bits), after that he will be able to convert them back to the original image. The second problem is that the tariffs for phone calls are too expensive and dictate each byte as 8 binary digits will last too long. To reduce costs, Alice and Bob agree to use

a more efficient data transfer method by using a special alphabet, which replaces every “six digits” with one “letter”.

### Advanced Encryption Standard

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six time faster than triple DES.

A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.

The features of AES are as follows –

- Symmetric key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger and faster than Triple-DES
- Provide full specification and design details
- Software implementable in C and Java

### Operation of AES:

AES is an iterative rather than Feistel cipher. It is based on ‘substitution–permutation network’. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix –

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

## **CHAPTER 4**

### **4 Project Purpose and Scope**

#### **4.1 Purpose**

The main purpose of this project is to monitor the student's attention on their school activities and to make the remote learning methods more effective.

#### **4.2 Project Scope**

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. To analyze the degree of attention of the students of each instance, it needs to be trained using machine learning algorithms. By using this algorithm, we can monitor the activities of the students and provides results to their parents based on the degree of attention of each and every student.

#### **4.3 Product Perspective**

We propose a smart way to proceed the remote learning method more effective by calculating students attention on their given activities and by motivating the students as well as the staffs and the parents.

#### **4.4 System Features**

This system makes the remote learning methods more effective and helpful for staffs students and parents. In this method; the students will perform in their exams, activities and in their online classes with more attention than before.

## **4.5 Design and Implementation Constraints**

### ***4.5.1 Constraints in Analysis***

- Constraints as Informal Text
- Constraints as Operational Restrictions
- Constraints Integrated in Existing Model
- Concepts Constraints as a Separate Concept
- Constraints Implied by the Model Structure

### ***4.5.2 Constraints in Design***

- Determination of the Involved Classes
- Determination of the Involved Objects
- Determination of the Involved Actions
- Determination of the Require Clauses
- Global actions and Constraint Realization

### ***4.5.3 Constraints in Implementation***

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object oriented methods.

## **4.6 Other Nonfunctional Requirements**

### ***4.6.1 Performance Requirements***

The application at this side controls and communicates with the following three main general components.

embedded browser in charge of the navigation and accessing to the web service;

Server Tier: The server side contains the main parts of the functionality of the proposed architecture. The components at this tier are the following.

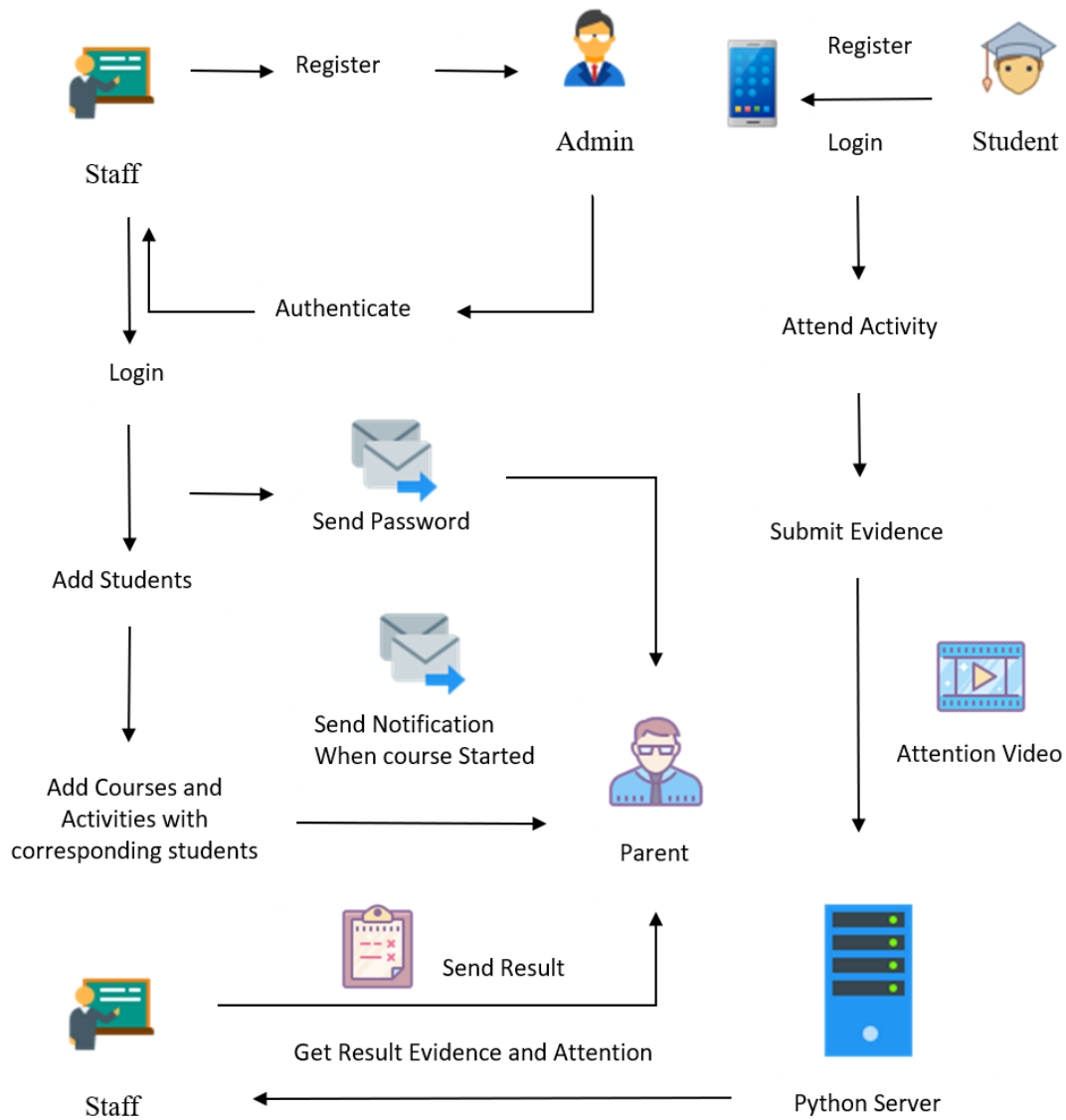
Web Server, Security Module, Server-Side Capturing Engine, Preprocessing Engine, Database System, Verification Engine, Output Module.

### ***4.6.2 Safety Requirements***

- The software may be safety-critical. If so, there are issues associated with its integrity
- The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.
- If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.
- There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable.
- If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.
- Systems with different requirements for safety levels must be separated.
- Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

## CHAPTER 5

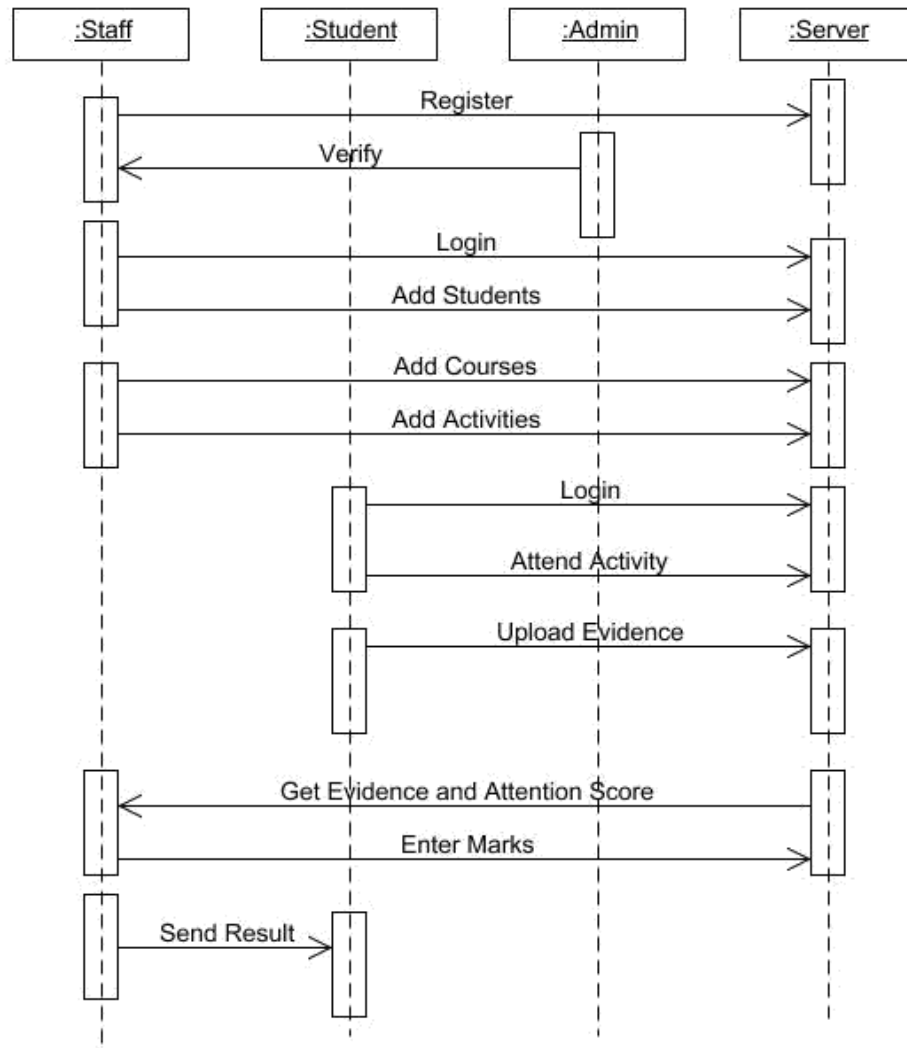
### 5.1 Architecture Diagram:



**Fig: 5.1**

## 5.2 Sequence Diagram:

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.



**Fig: 5.2**

### 5.3 Use Case Diagram:

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

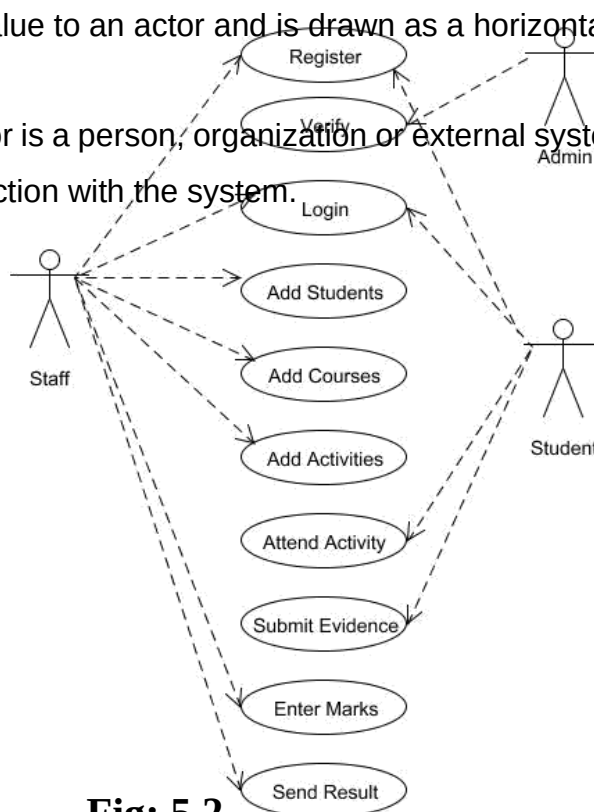
#### 5.3.1. USECASE DIAGRAM

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

**Use case:** A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

**Actor:** An actor is a person, organization or external system that plays a role in one or more interaction with the system.



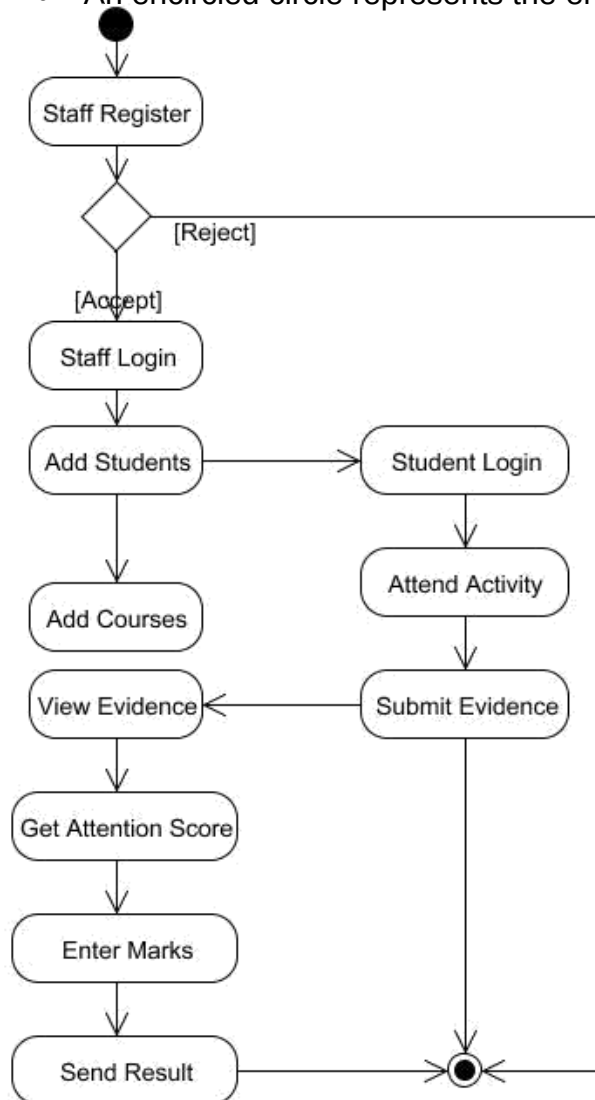
**Fig: 5.2**

## 5.4 Activity Diagram:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

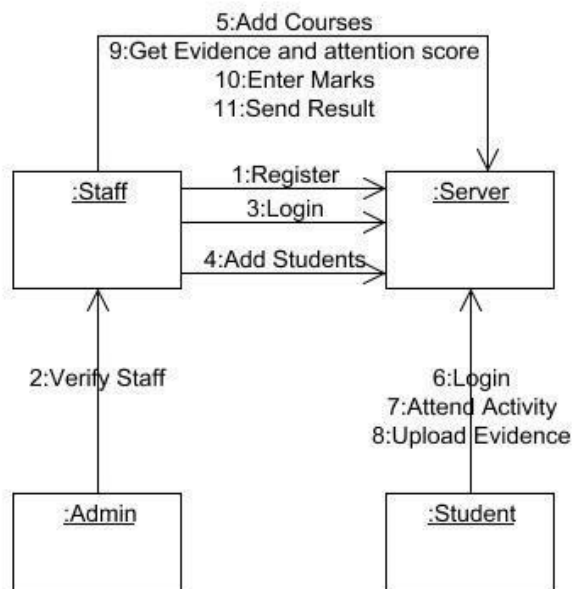
- Rounded rectangles represent activities
- Diamonds represent decisions
- Bars represent the start or end of concurrent activities
- A black circle represents the start of the workflow
- An encircled circle represents the end of the workflow



**Fig: 5.3**

## 5.5 Collaboration Diagram:

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.

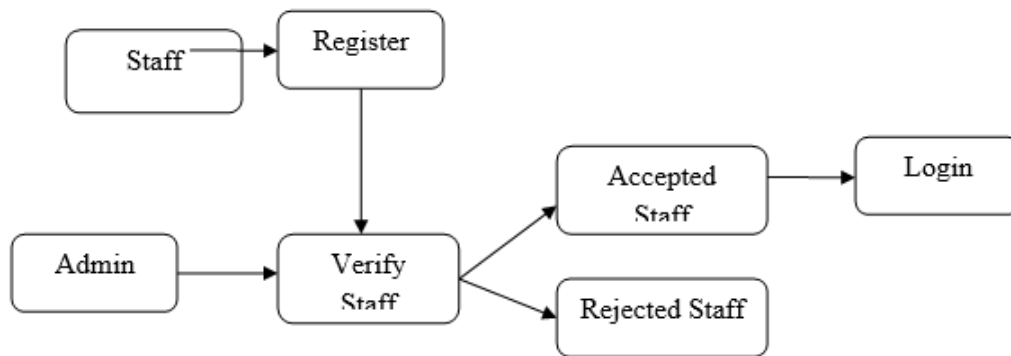


**Fig: 5.4**

## 5.6 DATA FLOW DIAGRAM:

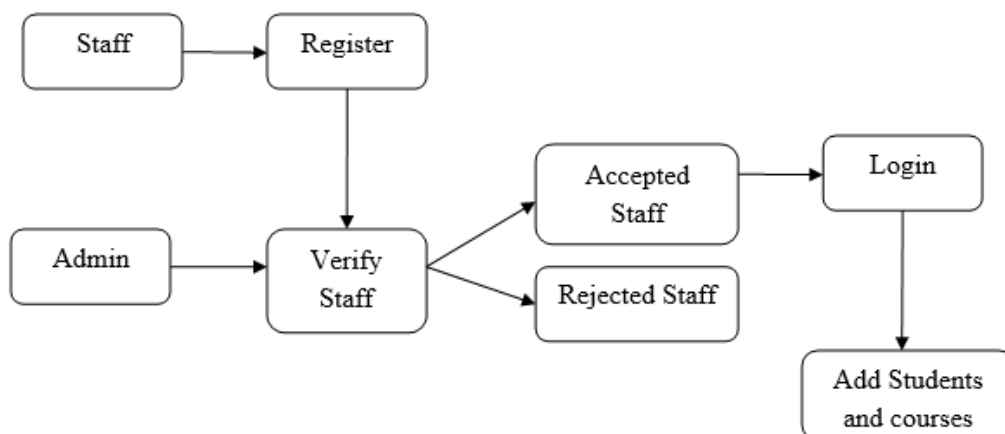
A Data Flow Diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its aspects. It is a preliminary step used to create an overview of the system which can later be elaborated DFDs can also be used for visualization of data processing.

### Level 0:



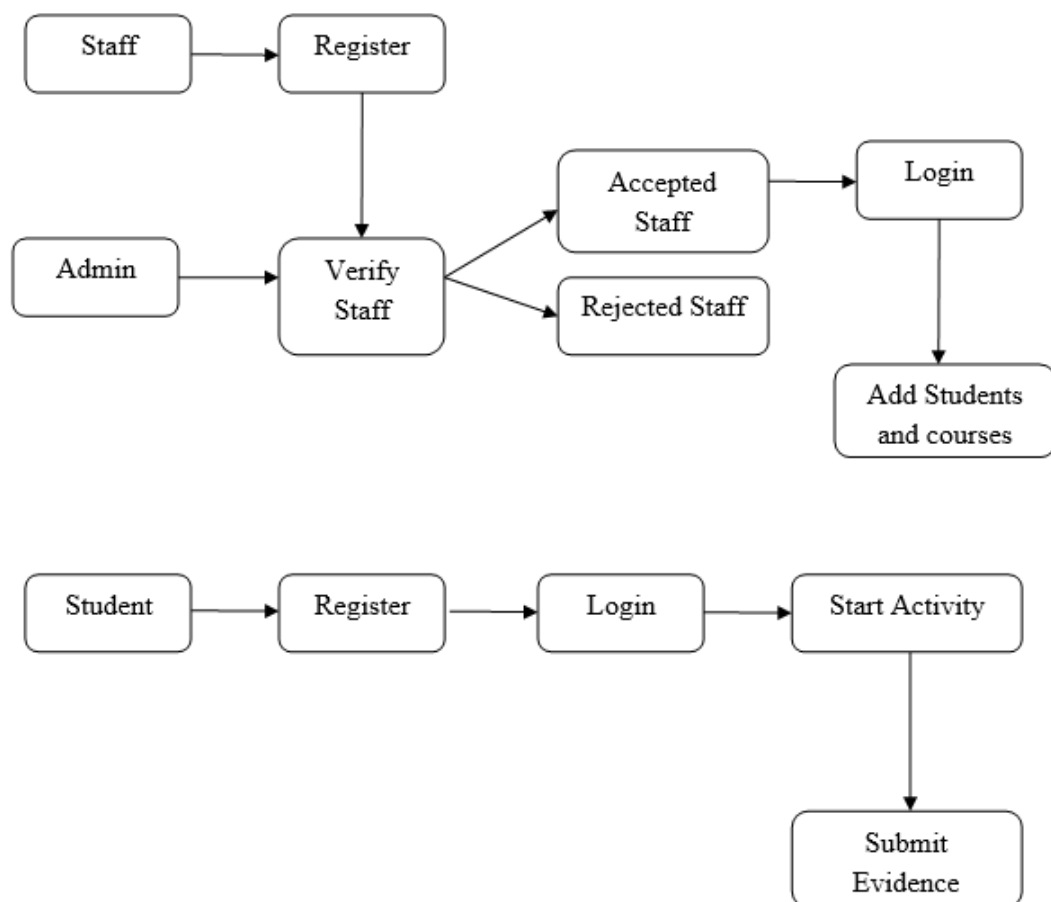
**Fig: 5.5**

### Level 1:



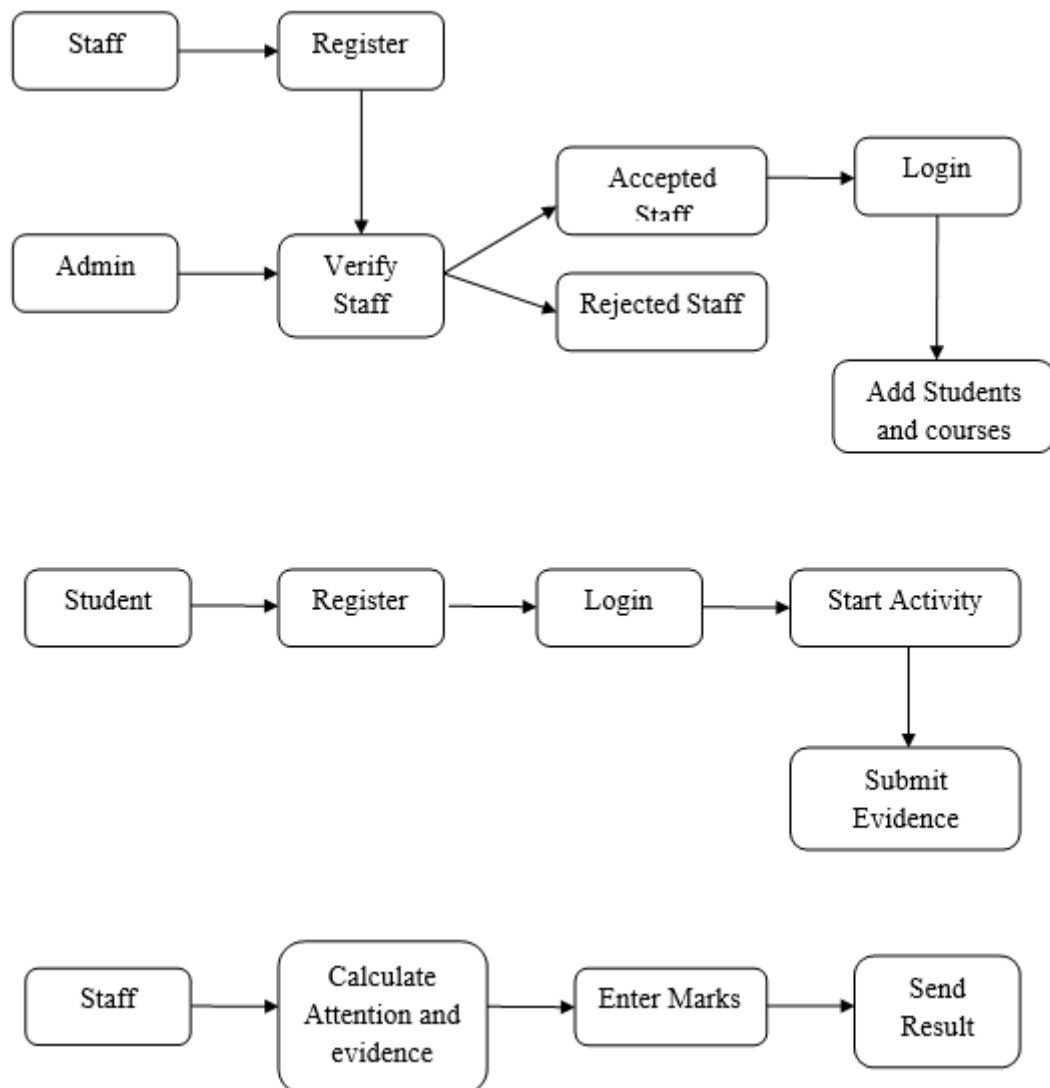
**Fig: 5.6**

**Level 2:**



**Fig: 5.7**

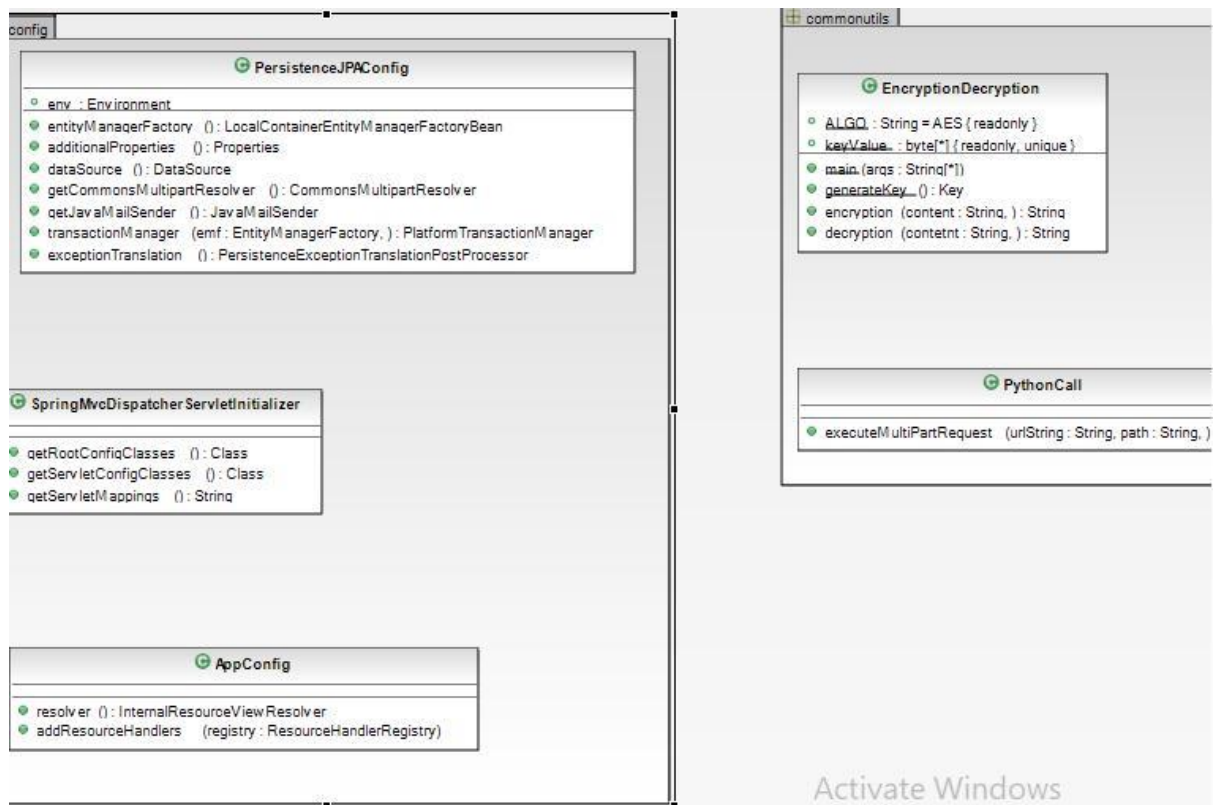
### Level 3:



**Fig: 5.8**

## 5.7 Class Diagram:

A Class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.



**Fig: 5.9**

## **CHAPTER 6**

### **SYSTEM DESIGN**

In this system, we propose a remote learning system and make it more efficient and more useful for parents, students and staffs. The staff will register their details. The Admin have to give permission for the registered staff to proceed with their login process. If the admin reject a staff, then that staff will not be accessible with their further process. The authorized staffs will proceeds their process after logging in. They will add the students along with their personal details who are under his/her class. Then the staff will upload tasks or activities under courses with any number of selected students who need to attend those activities. The students will register with their android mobile phone. Once the course gets started, mail will be sent through their parent's email. After getting mail, students will login using their registered email and the password which gets generated while staffs adding students and it will be sent through the parent's email and the student will start the activity. After completing every activity, the image-based evidence and the video of student while doing activity (attention video) will be saved. The staffs can view the created courses and activities. The students who are added in the activity will get displayed inside that activity. For every activity, the staff can view the performance of the student i.e., the image-based evidence after their activity completion. The attention will get calculated using the attention video in python. After viewing the evidence and the percentage of attention, the staff will enter the mark based on their evidence and attention. Finally, the result will be sent to the parents.

### **6.1MODULES**

- Staff Authentication
- Add students and courses
- Student Registration and Attention Detection
- Sending Result to Parents

## **6.2 MODULE EXPLANATION:**

### **6.2.1 *Staff Authentication***

The staff will register with their details. The Admin have to give permission for the registered staff to proceed with their login process. If the admin reject a staff, then that staff will not be accessible with their further process.

### **6.2.2 *Add students and courses***

The authorised staff will proceed their process after logging in. They will add the students along with their personal details who are in his/her class. Then the staff will upload tasks or activities under courses with any number of selected students who need to attend those activities.

### **6.2.3 *Student Registration and Attention Detection***

The students will register with their android mobile phone. Once the course get started, mail will be sent through their parent's email. After getting mail, students will login using their registered email and password and start the activity. After completing every activity, the image-based evidence and the attention video during their activity will be saved.

### **6.2.4 *Sending Result to Parents***

The staffs can view the created courses and activities. The students who are added in the activity will get displayed inside that activity. For every activity, the staff can view the performance of the student i.e., the image-based evidence after their activity completion. The attention will get calculated using the attention video in python. After viewing the evidence and the percentage of attention, the staff will enter the mark based on their evidence and attention. Finally, the result will sent to the parents.

## CHAPTER 7

### CODING AND TESTING

#### 7.1 CODING

Once the design aspect of the system is finalized the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

#### 7.2 CODING STANDARDS

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

- Program should be simple, clear and easy to understand
- Naming conventions
- Value conventions
- Script and comment procedure
- Message box format
- Exception and error handling

##### 7.2.1 NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be **self-descriptive**. One should even get the meaning and scope of

the variable by its name. The conventions are adopted for **easy understanding** of the intended message by the user. So it is customary to follow the conventions. These conventions are as follows:

### **Class names**

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

### **Member Function and Data Member name**

Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in lowercase.

## **7.2.2 VALUE CONVENTIONS**

Value conventions ensure values for variable at any point of time. This involves the following:

- Proper default values for the variables
- Proper validation of values in the field
- Proper documentation of flag values

## **7.2.3 SCRIPT WRITING AND COMMENTING STANDARD**

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

## **7.2.4 MESSAGE BOX FORMAT**

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

- X- User has performed illegal operation
- !- Information to the user

## **7.3 TEST PROCEDURE**

### **SYSTEM TESTING**

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source Code
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data

## **7.4 TEST DATA AND OUTPUT**

### **7.4.1 UNIT TESTING**

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

### **7.4.2 FUNCTIONAL TESTS**

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and

special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test

#### **7.4.3 PERFORMANCE TEST**

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

#### **7.4.4 STRESS TEST**

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

#### **7.4.5 STRUCTURED TEST**

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

- Exercise all logical decisions on their true or false sides
- Execute all loops at their boundaries and within their operational bounds
- Exercise internal data structures to assure their validity
- Checking attributes for their correctness
- Handling end of file condition, I/O errors, buffer problems and textual errors in output information

#### **7.4.6 INTEGRATION TESTING**

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

### **7.5 TESTING TECHNIQUES / TESTING STRATEGIES**

#### **7.5.1 TESTING**

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is

important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

#### *7.5.1.1 WHITE BOX TESTING*

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices control

#### *7.5.1.2 BLACK BOX TESTING*

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been

adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

### **7.5.2 SOFTWARE TESTING STRATEGIES:**

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works “outward” toward the integration of the entire computer based system
- Different testing techniques are appropriate at different points in time
- The developer of the software and an independent test group conducts testing
- Testing and Debugging are different activities but debugging must be accommodated in any testing strategy

#### **7.5.2.1 INTEGRATION TESTING:**

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem of course, is “putting them together”- interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function;

individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

#### *7.5.2.2 PROGRAM TESTING:*

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

#### *7.5.2.3 SECURITY TESTING:*

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

#### *7.5.2.4 VALIDATION TESTING*

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final

series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

- The function or performance characteristics confirm to specifications and are accepted
- A validation from specification is uncovered and a deficiency created

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic.

#### *7.5.2.5 USER ACCEPTANCE TESTING*

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

- Input screen design
- Output screen design

## SOURCE CODE

```
package com.staff.controller;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.PrintWriter;
import java.net.URL;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Random;

import javax.mail.internet.MimeMessage;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
import org.json.simple.parser.JSONParser;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.PropertySource;
import org.springframework.mail.SimpleMailMessage;
import org.springframework.mail.javamail.JavaMailSender;
import org.springframework.mail.javamail.MimeMessageHelper;
import org.springframework.mail.javamail.MimeMessagePreparator;
```

```
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.multipart.commons.CommonsMultipartFile;
import org.springframework.web.servlet.ModelAndView;
```

```
import com.staff.commonutils.EncryptionDecryption;
import com.staff.commonutils.PythonCall;
import com.staff.entity.Activity;
import com.staff.entity.Courses;
import com.staff.entity.FileUpload;
import com.staff.entity.Staff;
import com.staff.entity.Student;
import com.staff.entity.StudentActivity;
import com.staff.service.CustomerService;
```

```
@Controller
```

```
@RequestMapping
```

```
@PropertySource({ "classpath:mail.properties" })
```

```
@PropertySource({ "classpath:filelocation.properties" })
```

```
public class StaffController {
    public HashMap<String, String> map = new HashMap<String, String>();
    @Value("${imagepath}")
    private String imgpath;
    @Value("${activitypath}")
    private String actpath;
    @Value("${resultpath}")
    private String respath;
```

```
@Value("${user}")
private String user;

@Autowired
private JavaMailSender mailSenderObj;
int i = 0;
```

```
@Autowired
private CustomerService customerService;
```

```
@GetMapping("/")
public String index() {
    return "index";
}
```

```
@GetMapping("/index")
public String index1() {
    return "index";
}
```

```
@GetMapping("/staffview")
public String staffView() {
    return "staffview";
}
```

```
@GetMapping("/addstudent")
public String addStudent() {
    return "addstudent";
}
```

```
@GetMapping("/actview/getstudforact/getresult/{listvalue.evidence}")
public String getResult(Model m, @PathVariable("listvalue.evidence") String
filename) {
```

```

        String ext = filename.split("\\.")[1];
        m.addAttribute("filename", filename);
        m.addAttribute("ext", ext);
        return "getresult";
    }

```

```

    @GetMapping("/actview/getactfile/{listvalue.affiliate}")
    public String getActFile(Model m, @PathVariable("listvalue.affiliate") String
affiliate) {
        String ext = affiliate.split("\\.")[1];
        m.addAttribute("filename", affiliate);
        m.addAttribute("ext", ext);
        return "getactfile";
    }

```

```

    @GetMapping("/editstudent")
    public String editStudent(Model theModel) {
        Student stud = new Student();
        List<Student> studentList = new ArrayList<Student>();
        studentList = customerService.getStudentList();
        theModel.addAttribute("studentlist", studentList);
        if (studentList == null) {
            theModel.addAttribute("id", studentList.get(0).getId());
            theModel.addAttribute("sname",
studentList.get(0).getSname());
            theModel.addAttribute("semail",
studentList.get(0).getSemail());
            theModel.addAttribute("sdept", studentList.get(0).getSdept());
            theModel.addAttribute("saddress",
studentList.get(0).getSaddress());
            theModel.addAttribute("scontact",
studentList.get(0).getScontact());

```

```

        theModel.addAttribute("imgfile",
studentList.get(0).getImgfile());
    }
    return "editstudent";
}

```

```

@GetMapping("/showRegForm")
public String showFormForReg(Model theModel) {
    Staff staff = new Staff();
    theModel.addAttribute("customer", staff);
    return "register";
}

```

```

@GetMapping("/showLogForm")
public String showFormForLog(Model theModel) {
    Staff staff = new Staff();
    theModel.addAttribute("customer", staff);
    return "login";
}

```

```

@GetMapping("/courses")
public String viewCourses() {
    return "courses";
}

```

```

@GetMapping("/activity")
public String addStudent(Model theModel) {
    Student stud = new Student();
    List<Student> studentList = new ArrayList<Student>();
    studentList = customerService.getStudentList();
    theModel.addAttribute("studentlist", studentList);
    if (studentList == null) {

```

```

        theModel.addAttribute("id", studentList.get(0).getId());
        theModel.addAttribute("sname",
studentList.get(0).getSname());
        theModel.addAttribute("semail",
studentList.get(0).getSemail());
        theModel.addAttribute("sdept", studentList.get(0).getSdept());
        theModel.addAttribute("saddress",
studentList.get(0).getSaddress());
        theModel.addAttribute("scontact",
studentList.get(0).getScontact());
        theModel.addAttribute("imgfile",
studentList.get(0).getImgfile());
    }
    return "addcourses";
}

```

```

@GetMapping("/viewcourse")
public String viewCourses(Model theModel) {
    List<Courses> courseList = new ArrayList<Courses>();
    courseList = customerService.getCourseList();
    theModel.addAttribute("courselist", courseList);
    if (courseList == null) {
        theModel.addAttribute("cid", courseList.get(0).getId());
        theModel.addAttribute("cname",
courseList.get(0).getCname());
        theModel.addAttribute("cdesc", courseList.get(0).getCdesc());
        theModel.addAttribute("cnum", courseList.get(0).getCnum());
        theModel.addAttribute("cstatus",
courseList.get(0).getCstatus());
    }
    return "courses";
}

```







```

//      @GetMapping("/actview/activityfile/${listvalue.aname}")
//      public      ModelAndView      actFile(ModelAndView      mv,
@RequestParam("listvalue.aname") String aname) {
//          String actFile = customerService.actFile(aname);
//          System.out.println("----->>>>" + actFile);
//          try {
//              File file=new File("E:\\Apache\\" + actFile);
//              FileInputStream fis=new FileInputStream(file);    //opens a connection
to an actual file
//              System.out.println("file content: ");
//              int r=0;
//              while((r=fis.read())!=-1)
//              {
//                  System.out.print((char)r);    //prints the content of the file
//              }
//          }
//          catch(Exception e)
//          {
//              e.printStackTrace();
//          }
//          mv.setViewName("actview");
//          return mv;
//      }

```

```

        @PostMapping("/saveStaff")
        public      ModelAndView      saveCustomer(ModelAndView      mv,
@RequestParam("name") String name,
                @RequestParam("email")      String      email,
@RequestParam("phone") String phone,
                @RequestParam("address")      String      address,
@RequestParam("subject") String subject,
                @RequestParam("password") String pass) {

```

```

        Staff staff = new Staff();
        staff.setName(name);
        staff.setEmail(email);
        staff.setPhone(phone);
        staff.setAddress(address);
        staff.setSubject(subject);
        staff.setPass(pass);
        System.out.println("-----=>>>" + name + "-----=>>>" + email + "-----
=>>>" + address + "-----=>>>" + phone
                        + "-----=>>>" + subject + "-----=>>>" + pass);
        String check = customerService.checkStaff(staff);
        if (check.equalsIgnoreCase("0")) {
            int count = customerService.saveStaff(staff);
            if (count > 0) {
                mv.addObject("msg", "Registration Successfull");
                mv.setViewName("login");
            } else {
                mv.addObject("msg", "Registration Failed");
                mv.setViewName("register");
            }
        } else {
            mv.addObject("msg", "Staff Already Registered");
            mv.setViewName("register");
        }
        return mv;
//        customerService.saveCustomer(theCustomer);
//        return "redirect:/customer/";
    }

```

```

    @PostMapping("/login")
    public ModelAndView staffLogin(ModelAndView mv, HttpSession sess,
    @RequestParam("email") String email,

```

```

        @RequestParam("password") String pass) {
    Staff staff = new Staff();
    staff.setEmail(email);
    staff.setPass(pass);
    System.out.println("=----=>>>" + email + "=----=>>>" + pass);
    String count = customerService.staffLogin(staff);
    if (count == null) {
        mv.addObject("msg", "Login Failed...Invalid Details or Wait for
admin to give Permission... ");
        mv.setViewName("login");
    } else {
        mv.addObject("msg", "Login Successfull...");
        sess.setAttribute("usermail", email);
        mv.addObject("usermail", email);
        mv.setViewName("staffview");
    }
    return mv;
}

```

```

@PostMapping("/addstud")
public ModelAndView addStud(ModelAndView mv,
@RequestParam("imgfile") CommonsMultipartFile imgfile,
@RequestParam("semail") final String semail,
@RequestParam("sname") String sname,
@RequestParam("sclass") String sdept,
@RequestParam("saddress") String saddress,
@RequestParam("scontact") String scontact) throws
IOException {
    String otp = null;
    String filename = imgfile.getOriginalFilename();
    byte[] bytes = imgfile.getBytes();
    // String fileData = new String(bytes);
}

```

```

//      System.out.println("-----" + bytes + "=====" + fileData);
//      String type = imgfile.getContentType();
      Student stud = new Student();
      stud.setSname(sname);
      stud.setSemail(semicolon);
      stud.setSdept(sdept);
      stud.setSaddress(saddress);
      stud.setScontact(scontact);
      stud.setImgfile(filename);
      File f = new File(imgpath + filename);
      FileOutputStream fos = new FileOutputStream(f);
      fos.write(bytes);
      fos.close();
      String check = customerService.checkStud(stud);
      if (check.equalsIgnoreCase("0")) {
          try {
              Random r = new Random();

              char[] map1 = new char[68];
              int i = 0;
              for (char c = 'A'; c <= 'Z'; c++) {
                  map1[i++] = c;
              }
              for (char c = 'a'; c <= 'z'; c++) {
                  map1[i++] = c;
              }
              for (char c = '0'; c <= '9'; c++) {
                  map1[i++] = c;
              }
              map1[i++] = '+';
              map1[i++] = '/';
              map1[i++] = '$';
          }
      }
  }
}

```

```

        map1[i++] = '#';
        map1[i++] = '&';
        map1[i++] = '*';
        String p1 = String.valueOf(map1[r.nextInt(67)]);
        String p2 = String.valueOf(map1[r.nextInt(67)]);
        String p3 = String.valueOf(map1[r.nextInt(67)]);
        String p4 = String.valueOf(map1[r.nextInt(67)]);
        String p5 = String.valueOf(map1[r.nextInt(67)]);
        String p6 = String.valueOf(map1[r.nextInt(67)]);
        String p7 = String.valueOf(map1[r.nextInt(67)]);
        String p8 = String.valueOf(map1[r.nextInt(67)]);
        otp = p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8;
        final String message = "Your child " + sname + "'s Login
Password is :" + otp + ""';
        final String subject = "Online Class Login Password";
        mailSenderObj.send(new MimeMessagePreparator() {
            public void prepare(MimeMessage
mimeMessage) throws Exception {
                MimeMessageHelper mimeMsgHelperObj
= new MimeMessageHelper(mimeMessage, true, "UTF-8");
                mimeMsgHelperObj.setTo(semail);
                mimeMsgHelperObj.setFrom(user);
                mimeMsgHelperObj.setText(message);
                mimeMsgHelperObj.setSubject(subject);
            }
        });
    } catch (Exception e) {
        e.printStackTrace();
    }
    stud.setPassword(otp);
    int count = customerService.saveStud(stud);
    if (count > 0) {

```

```

        mv.addObject("msge", "Added Successfully");
        mv.setViewName("addstudent");
    } else {
        mv.addObject("msge", "Failed");
        mv.setViewName("addstudent");
    }
} else {
    mv.addObject("msge", "Student Already Added");
    mv.setViewName("addstudent");
}
return mv;
}

```

```

@PostMapping("/deletestud")
public ModelAndView deleteStud(ModelAndView mv,
@RequestParam("data") String mail, HttpServletRequest request,
    HttpServletResponse response) throws IOException {
    PrintWriter printWriter = response.getWriter();
    int count = customerService.deleteStud(mail);
    if (count > 0) {
        mv.addObject("msge", "Removed Successfully");
        printWriter.print(1);
        mv.addObject("resultData", 1);
        mv.setViewName("addstudent");
    } else {
        mv.addObject("msge", "Failed");
        mv.setViewName("addstudent");
    }
    return mv;
}

```

```

    @GetMapping("/deletecourse")
    public ModelAndView deleteCourse(ModelAndView mv,
    @RequestParam("data") String mail, HttpServletRequest request,
    HttpServletResponse response) throws IOException {
        PrintWriter printWriter = response.getWriter();
        int count = customerService.deleteCourse(mail);
        if (count > 0) {
            mv.addObject("msg", "Removed Successfully");
            printWriter.print(1);
            mv.addObject("resultData", 1);
            mv.setViewName("courses");
        } else {
            mv.addObject("msg", "Failed");
            mv.setViewName("courses");
        }
        return mv;
    }

```

```

    @GetMapping("/actview/deleteact")
    public ModelAndView deleteAct(ModelAndView mv,
    @RequestParam("data") String mail, HttpServletRequest request,
    HttpServletResponse response) throws IOException {
        PrintWriter printWriter = response.getWriter();
        int count = customerService.deleteAct(mail);
        System.out.println("=====count===== " + count + "--
---" + mail);
        if (count > 0) {
            mv.addObject("msg", "Removed Successfully");
            printWriter.print(1);
            mv.addObject("resultData", 1);
            mv.setViewName("actview");
        }
    }

```

```

    } else {
        mv.addObject("msg", "Failed");
        mv.setViewName("actview");
    }
    return mv;
}

@GetMapping("/actview/getstudforact/{listvalue.aname}")
public ModelAndView getstudforAct(ModelAndView mv, Model model,
HttpSession sess,
    @PathVariable("listvalue.aname") String aname) {
    System.out.println("=====-----aname-----===== " +
aname);
    String acid = customerService.courid(aname);
    sess.setAttribute("aname", aname);
    sess.setAttribute("activityid", acid);
    List<StudentActivity> stuid = customerService.getsid(acid);
    model.addAttribute("studentlist", stuid);
    if (stuid == null) {
        model.addAttribute("id", stuid.get(0).getId());
        model.addAttribute("studentname",
stuid.get(0).getStudentname());
        model.addAttribute("studentemail",
stuid.get(0).getStudentemail());
        model.addAttribute("evidence", stuid.get(0).getEvidence());
        model.addAttribute("marks", stuid.get(0).getMarks());
        model.addAttribute("attention", stuid.get(0).getAttention());
        model.addAttribute("status", stuid.get(0).getStatus());
    }
    mv.setViewName("studentactivity");
    return mv;
}

```

```

    @GetMapping("/actview/getstudforact/savemark")
    public ModelAndView saveMark(ModelAndView mv, Model model,
        HttpSession sess, HttpServletResponse response,
            @RequestParam("marks") String mark, @RequestParam("id")
String id) throws IOException {
        PrintWriter printWriter = response.getWriter();
        System.out.println("=====mark===== " + mark + "-----id-----"
+ id);

        int savemark = customerService.saveMarks(id, mark);
        String aname = (String) sess.getAttribute("aname");
        String acid = customerService.courid(aname);
        List<StudentActivity> stuid = customerService.getsid(acid);
        model.addAttribute("studentlist", stuid);
        if (stuid == null) {
            model.addAttribute("id", stuid.get(0).getId());
            model.addAttribute("studentname",
stuid.get(0).getStudentname());
            model.addAttribute("studentemail",
stuid.get(0).getStudentemail());
            model.addAttribute("evidence", stuid.get(0).getEvidence());
            model.addAttribute("marks", stuid.get(0).getMarks());
            model.addAttribute("attention", stuid.get(0).getAttention());
            model.addAttribute("status", stuid.get(0).getStatus());
        }
        mv.setViewName("studentactivity");
        return mv;
    }

```

```

    @PostMapping("/addcourse")
    public ModelAndView addCourse(ModelAndView mv,
        @RequestParam("cname") String cname,

```

```

        @RequestParam("cdesc")        String        cdesc,
@RequestParam("cnum") String cnum,
        @RequestParam("cclass")        String        cclass,
@RequestParam("cstudents") String cstudents, HttpSession sess) {
    Courses courses = new Courses();
    courses.setCname(cname);
    courses.setCdesc(cdesc);
    courses.setCnum(cnum);
    courses.setCclass(cclass);
    courses.setCstudents(cstudents);
    System.out.println("=----=>>>" + cname + "=----=>>>" + cdesc + "=----
=>>>" + cnum);
    String check = customerService.checkCourse(courses);
    if (check.equalsIgnoreCase("0")) {
        int count = customerService.saveCourse(courses);
        String cid = customerService.cid(courses);
        if (count > 0) {
            mv.addObject("m", "Course Created Successfully");
//            mv.addObject("cid", cid);
            sess.setAttribute("cnum", cnum);
            sess.setAttribute("cid", cid);
            mv.setViewName("activities");
        } else {
            mv.addObject("m", "Course Creation Failed");
            mv.setViewName("addcourses");
        }
    } else {
        mv.addObject("m", "Course Already Added");
        mv.setViewName("addcourses");
    }
    return mv;
}

```

```

    @GetMapping("/actview/getstudforact/getattention/{listvalue.id}/{listvalue.st
udentemail}")
    public ModelAndView getAttention(ModelAndView mv, Model model,
HttpSession sess,
        @PathVariable("listvalue.id") String id,
        @PathVariable("listvalue.studentemail") String studentemail) throws Exception {
        System.out.println("-----");
        String studentid = customerService.getEma(studentemail);
        String courseid = (String)sess.getAttribute("courseid");
        String activityid = (String)sess.getAttribute("activityid");

        System.out.println("*****"+activityid+"*****"+courseid+"*****"+
studentid);

        String getValue =
customerService.getVal(courseid,activityid,studentid);
        System.out.println("-----Attention Path-----"+getValue);
        PythonCall pc = new PythonCall();
        List<String> respo =
pc.executeMultiPartRequest("http://localhost:5001/register", getValue);
        String res = respo.toString();
        System.out.println("-----respo-----" + res);
        JSONArray jaArray = (JSONArray) new JSONParser().parse(res);
        JSONObject jsonObject = (JSONObject) jaArray.get(0);
        String data = String.valueOf(jsonObject.get("attention"));
        System.out.println("-----data&id-----" + data + "-----" + id);
        int saveatt = customerService.saveAttention(id, data);
        String aname = (String) sess.getAttribute("aname");
        String acid = customerService.courid(aname);
        List<StudentActivity> stuid = customerService.getsid(acid);
        model.addAttribute("studentlist", stuid);
        if (stuid == null) {

```

```

        model.addAttribute("id", stuid.get(0).getId());
        model.addAttribute("studentname",
stuid.get(0).getStudentname());
        model.addAttribute("studentemail",
stuid.get(0).getStudentemail());
        model.addAttribute("evidence", stuid.get(0).getEvidence());
        model.addAttribute("marks", stuid.get(0).getMarks());
        model.addAttribute("attention", stuid.get(0).getAttention());
        model.addAttribute("status", stuid.get(0).getStatus());
    }
    mv.setViewName("studentactivity");
    return mv;
}

//      @PostMapping("/addact/{usermail}")
//      public      ModelAndView      addActivity(ModelAndView      mv,
//      @RequestParam("file") CommonsMultipartFile file,
//      @RequestParam("checkboxName") String[] checkboxValue,
//      HttpSession session,
//      @PathVariable("usermail")      String      usermail)      throws
//      IOException {
//      String      path1      =
"F:\\AbiramiWorkspace\\StaffPortal\\src\\main\\resources\\images\\ActivityFiles\\";
//
//      usermail = (String) session.getAttribute("usermail");
//      List list = Arrays.asList(checkboxValue);
//      String val = list.toString();
//      session.setAttribute("checked", list);
//      FileUpload upload = new FileUpload();
//      EncryptionDecryption      encryptionDecryption      =      new
EncryptionDecryption();
//      String filename = file.getOriginalFilename();

```

```

//      byte[] bytes = file.getBytes();
//      String fileData = new String(bytes);
//      String type = file.getContentType();
//      String encryptedData = encryptionDecryption.encryption(fileData);
//      System.out.println("======" + list);
//      System.out.println("======" + usermail);
//      System.out.println("======" + filename);
//      System.out.println("======" + type);
//      System.out.println("======" + encryptedData);
//
//      upload.setStudentnames(val);
//      upload.setFile(encryptedData);
//      upload.setFilename(filename);
//      upload.setFiletype(type);
//      upload.setUploadedby(usermail);
//      File f = new File(path1 + filename);
//
//      FileOutputStream fos = new FileOutputStream(f);
//      fos.write(bytes);
//      fos.close();
//      int count = customerService.saveFile(upload);
//      if (count > 0) {
//          mv.addObject("ms", "success");
//          mv.setViewName("activity");
//
//      } else {
//          mv.addObject("ms", "failed");
//          mv.setViewName("activity");
//
//      }
//      return mv;
//  }

```

```

    @PostMapping("/addactivity")
    public ModelAndView addActivity(ModelAndView mv, HttpSession sess,
        @RequestParam("afile") CommonsMultipartFile afile,
        @RequestParam("aname") String aname,
        @RequestParam("adesc") String adesc,
        @RequestParam("aduration") String aduration,
        @RequestParam("cid") String cid, @RequestParam("cnum")
        String cnum) throws IOException {
        StudentActivity studact = new StudentActivity();
        //      System.out.println("-----hitcounts====="+"hitcounts+"
        "+cnum);
        //      String      pathname      =
        "F:\\AbiramiWorkspace\\StaffPortal\\src\\main\\resources\\images\\ActivityFiles\\";
        //      String      pathname      =
        "http://localhost:7777/Admin/resources/images/ActivityFiles/";
        FileUpload upload = new FileUpload();
        EncryptionDecryption encryptionDecryption = new
        EncryptionDecryption();
        String filename1 = afile.getOriginalFilename();
        byte[] bytes1 = afile.getBytes();
        String fileData1 = new String(bytes1);
        String type = afile.getContentType();
        String encryptedData = encryptionDecryption.encrypted(fileData1);
        System.out.println("=====" + filename1);
        System.out.println("=====" + type);
        System.out.println("=====" + encryptedData);
        //      String
        path=sess.getServletContext().getRealPath("/resources/images/ActivityFiles/");
        //      System.out.println("----->" + path);
        upload.setFile(encryptedData);
    }

```

```

        upload.setFilename(filename1);
        upload.setFiletype(type);
        File f = new File(actpath + filename1);
        FileOutputStream fos1 = new FileOutputStream(f);
        fos1.write(bytes1);
        fos1.close();
//
        int count1 = customerService.saveFile(upload);
        String filename = afile.getOriginalFilename();
        byte[] bytes = afile.getBytes();
        String fileData = new String(bytes);
        Activity act = new Activity();
        act.setCid(cid);
        act.setAname(aname);
        act.setAdescip(adesc);
        act.setAduration(aduration);
        act.setAfile(filename);
        File file = new File(actpath + filename);
        FileOutputStream fos = new FileOutputStream(file);
        fos.write(bytes);
        fos.close();
        System.out.println("=----=>>>" + aname + "=----=>>>" + adesc + "=---
-=>>>" + aduration + "=----=>>>" + cid);
//
        if(Integer.parseInt(hitcounts) < Integer.parseInt(cnum)) {
            String check = customerService.checkAct(act);
            if (check.equalsIgnoreCase("0")) {
                int count = customerService.saveActivity(act);
                if (count > 0) {
                    mv.addObject("me", "Activity Created Successfully");
                    mv.setViewName("activities");
                    if (i < Integer.parseInt(cnum) - 1) {
                        i += 1;
                    } else {

```

```

        i = 0;
        String getstud = customerService.setCourse(cid);
        List<StudentActivity> getactid =
customerService.setStudAct(cid);
        List<String> getactids = new ArrayList<String>();
        String[] studsid;
        studsid = getstud.split(",");
        List<String> getstuids = Arrays.asList(studsid);
        for (int i = 0; i < getactid.size(); i++) {
            int ids = getactid.get(i).getId();
            String allid = Integer.toString(ids);
            getactids.add(allid);
        }
        System.out.println(",,,,,,,,,,,,,,,,,,,,,,,,,,,,," +
getstuids + ",,,,," + getactids);
        String finalSave = customerService.saveids(cid,
getstuids, getactids);
        System.out.println("=====-----
===== "+ finalSave);
        mv.addObject("comp", "Course Creation
Completed");
        mv.setViewName("addcourses");
    }
} else {
    mv.addObject("me", "Activity Creation Failed");
    mv.setViewName("addcourses");
}
} else {
    mv.addObject("me", "Activity Already Added");
    mv.setViewName("activities");
}
return mv;

```

```

    }

    @GetMapping("/getstudforclass")
    public void getstudforclass(ModelAndView mv, HttpServletRequest request,
    HttpServletResponse response,
        Model theModel, @RequestParam("data") String stuclass)
    throws IOException {
        PrintWriter printWriter = response.getWriter();
        //      System.out.println("=====stuclass=====>>>"      +
        stuclass);
        List<Student> studList = new ArrayList<Student>();
        studList = customerService.getstudList(stuclass);
        //      System.out.println("=====map===== " + studList.get(i).getId()
        + studList.get(i).getSname());
        Map<String, String> stuList = new HashMap<String, String>();
        //      int id = studList.get(i).getId();
        String stuid = Integer.toString(studList.get(i).getId());
        String sname = studList.get(i).getSname();
        for (int i = 0; i <= studList.size() - 1; i++) {
            stuList.put(studList.get(i).getSdept(), studList.get(i).getId() +
"@ " + studList.get(i).getSname());
        //      System.out.println("=====stuid=====>>>"      +
        stuid + sname);
            map.put(stuid, sname);
            String ar = stuList.toString();
            request.setAttribute("ar", ar);
            theModel.addAttribute("arr", studList);
        //      String arr = map.toString();
            printWriter.write(ar);
        }
        //      theModel.addAttribute("stuList", studList);
        //      mv.addObject("stuList", stuList);
    }

```

```

//          mv.setViewName("addcourses");
//          return mv;
//          return studList;
    }

    @PostMapping("/editstudents")
    public ModelAndView editStud(ModelAndView mv, Model theModel,
    @RequestParam("imgfile") CommonsMultipartFile imgfile,
        @RequestParam("semail") String semail,
    @RequestParam("sname") String sname,
        @RequestParam("sclass") String sdept,
    @RequestParam("saddress") String saddress,
        @RequestParam("scontact") String scontact) throws
IOException {
//          String path = "E://Apache//";
        String path1 =
"F:\\AbiramiWorkspace\\StaffPortal\\src\\main\\resources\\images\\StudentsId\\";
        String filename = imgfile.getOriginalFilename();
        byte[] bytes = imgfile.getBytes();
//          String fileData = new String(bytes);
//          System.out.println("-----" + bytes + "======" + fileData);
//          String type = imgfile.getContentType();
        Student stud = new Student();
        stud.setSname(sname);
        stud.setSemail(semail);
        stud.setSdept(sdept);
        stud.setSaddress(saddress);
        stud.setScontact(scontact);
        stud.setImgfile(filename);
        File f = new File(imgpath + filename);
        FileOutputStream fos = new FileOutputStream(f);
        fos.write(bytes);

```

```

        fos.close();
//        String check = customerService.checkStud(stud);
//        if (check.equalsIgnoreCase("0")) {
            int count = customerService.editStud(stud);
            if (count > 0) {
                List<Student> studentList = new ArrayList<Student>();
                studentList = customerService.getStudentList();
                theModel.addAttribute("studentlist", studentList);
                if (studentList == null) {
                    theModel.addAttribute("id", studentList.get(0).getId());
                    theModel.addAttribute("sname",
studentList.get(0).getSname());
                    theModel.addAttribute("semail",
studentList.get(0).getSemail());
                    theModel.addAttribute("sdept",
studentList.get(0).getSdept());
                    theModel.addAttribute("saddress",
studentList.get(0).getSaddress());
                    theModel.addAttribute("scontact",
studentList.get(0).getScontact());
                    theModel.addAttribute("imgfile",
studentList.get(0).getImgfile());
                }
                mv.addObject("msge", "Updated Successfully");
                mv.setViewName("editstudent");
            } else {
                List<Student> studentList = new ArrayList<Student>();
                studentList = customerService.getStudentList();
                theModel.addAttribute("studentlist", studentList);
                if (studentList == null) {
                    theModel.addAttribute("id", studentList.get(0).getId());

```

```

        theModel.addAttribute("sname",
studentList.get(0).getSname());
        theModel.addAttribute("semail",
studentList.get(0).getSemail());
        theModel.addAttribute("sdept",
studentList.get(0).getSdept());
        theModel.addAttribute("saddress",
studentList.get(0).getSaddress());
        theModel.addAttribute("scontact",
studentList.get(0).getScontact());
        theModel.addAttribute("imgfile",
studentList.get(0).getImgfile());
    }
    mv.addObject("msge", " Updation Failed");
    mv.setViewName("editstudent");
}
//    } else {
//        mv.addObject("msge", "Student Already Added");
//        mv.setViewName("addstudent");
//    }
    return mv;
}
}

```

## SNAPSHOTS

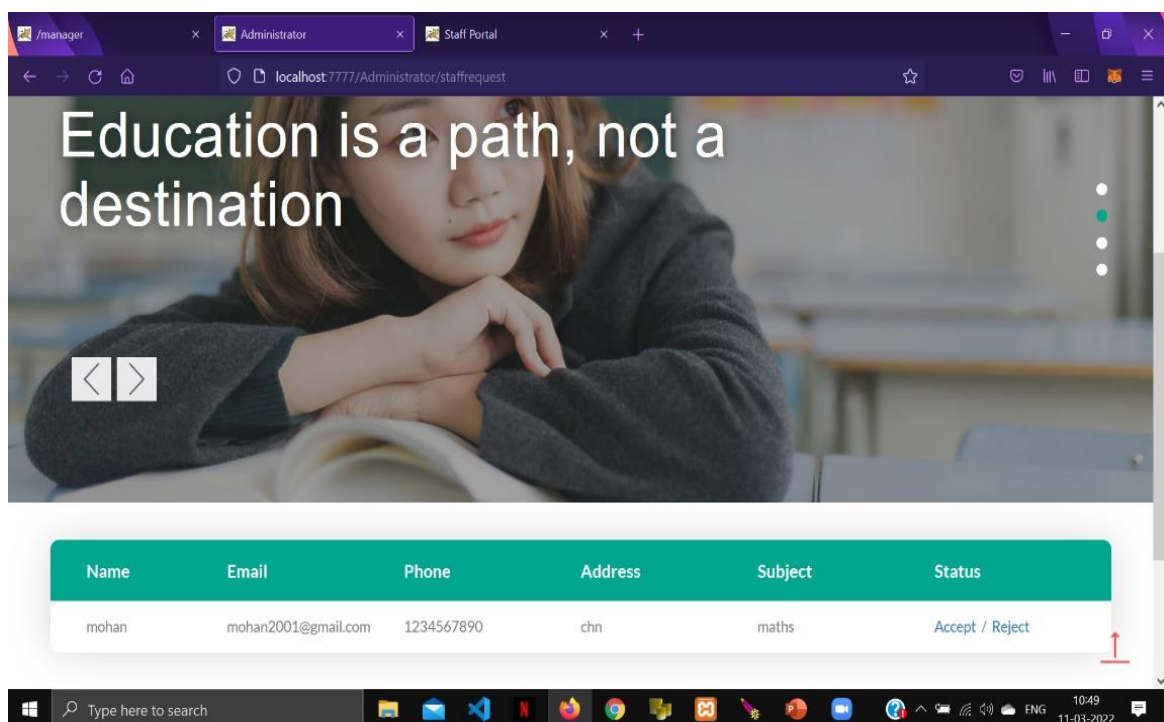
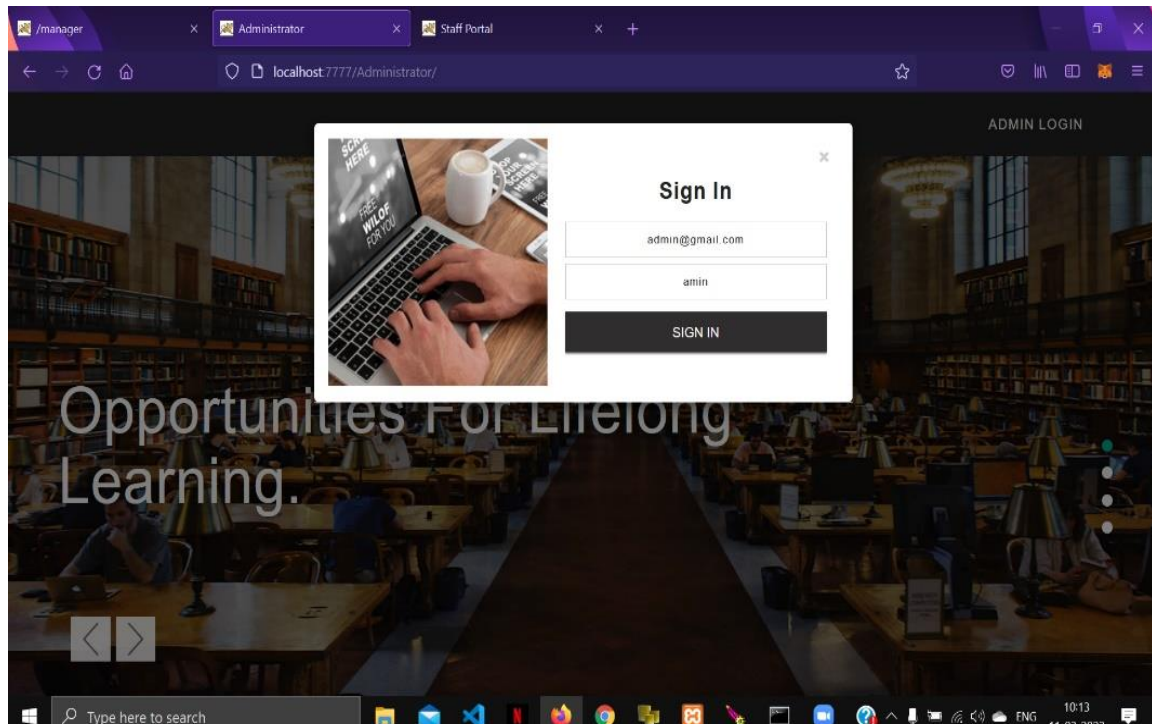
### REGISTER A STAFF DETAILS

The screenshot displays a web application interface for staff registration. The browser window shows three tabs: `/manager`, `Administrator`, and `Staff Portal`. The active tab is `Staff Portal`, and the address bar indicates the URL `localhost:7777/StaffPortal/showRegForm`. The page layout consists of a large red banner on the left side, featuring a man reading a book and the text "The Perfect Theme For Education And Training Center". On the right side, there is a registration form with the following fields:

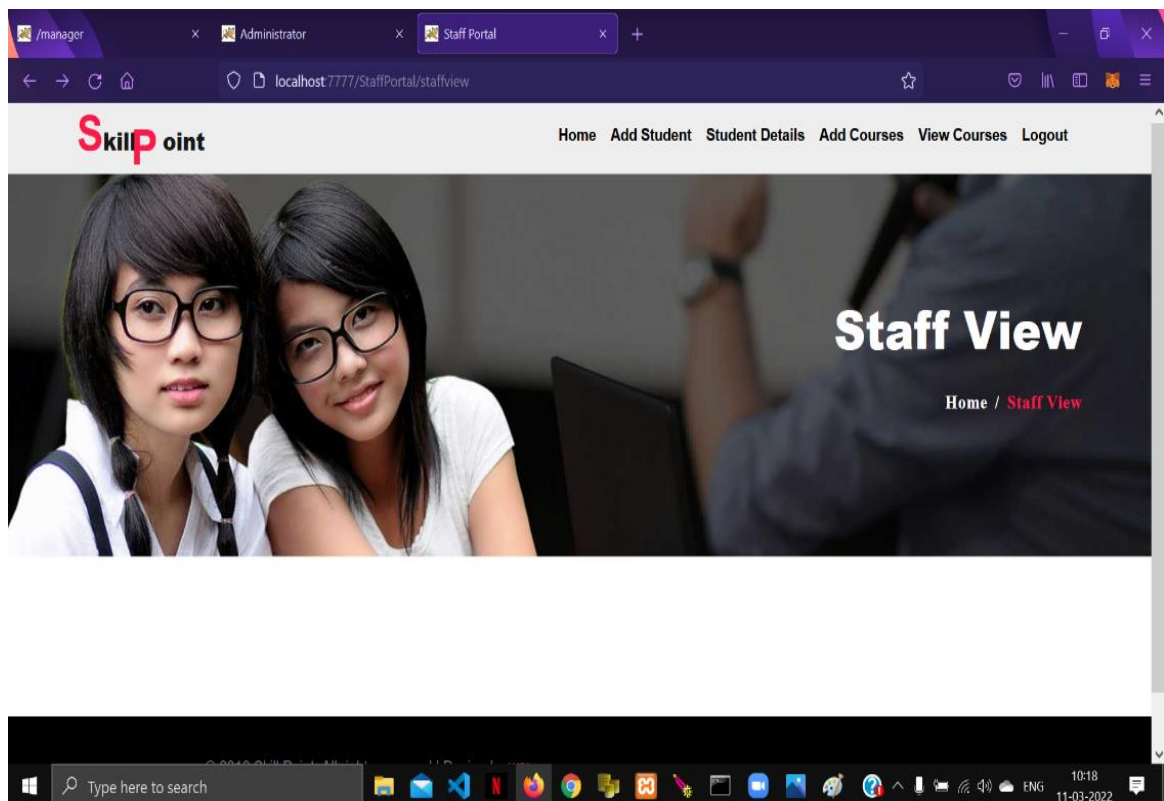
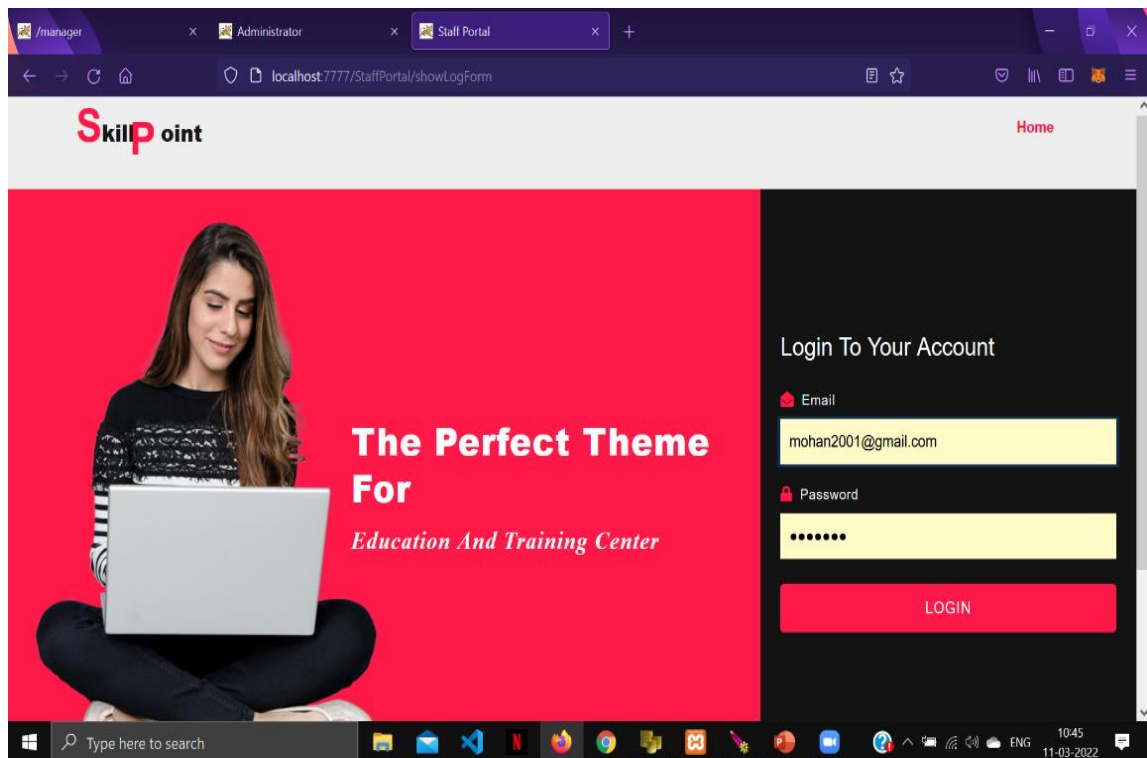
- Name**:
- Email**:
- Phone**:
- Address**:
- Subject**:
- Password**:

Below the form fields is a red button labeled **REGISTER**. The Windows taskbar is visible at the bottom of the screen, showing the search bar and various application icons.

## ADMIN PORTAL



## LOGIN STAFF PAGE



## ADD STUDENT DETAILS

**Add Student**

---

Add Student

Upload Student's Image

a1.jpg

## ADD COURSE DETAILS

The screenshot shows a web browser window with the URL `localhost:7777/StaffPortal/activity`. The page title is "Add Course". Below the title, there is a section labeled "Upload Course". This section contains five input fields, each with a red icon on the left:

- Field 1: Contains the text "tamil".
- Field 2: Contains the text "grammar".
- Field 3: Contains the text "1" and has a dropdown arrow on the right.
- Field 4: Contains the text "Class 1" and has a dropdown arrow on the right.
- Field 5: Contains the text "hari" and "mohan" (which is highlighted in blue), and has a dropdown arrow on the right.

Below the input fields is a red button labeled "Create Course". The Windows taskbar at the bottom shows the time as 10:20 on 11-03-2022.

The screenshot shows a web browser window with the URL `localhost:7777/StaffPortal/addcourse`. The page title is "Upload Activity". Below the title, there are four input fields, each with a red icon on the left:

- Field 1: Labeled "Enter Activity name" and contains the text "word".
- Field 2: Labeled "Enter Activity Description" and contains the text "list 5 types of grammer in tamil".
- Field 3: Labeled "Enter Activity Duration(in minutes)" and contains the text "5".
- Field 4: Labeled "Upload Activity" and contains a "Browse..." button followed by the text "a2.jpg".

Below the input fields is a red button labeled "Create Activity". The Windows taskbar at the bottom shows the time as 10:21 on 11-03-2022.

## COURSE START

The screenshot shows a web browser window with three tabs: `/manager`, `Administrator`, and `Staff Portal`. The address bar displays `localhost:7777/StaffPortal/viewcourse`. The main content area is divided into two columns, each representing a course.

**Course 1 (Left Column):**

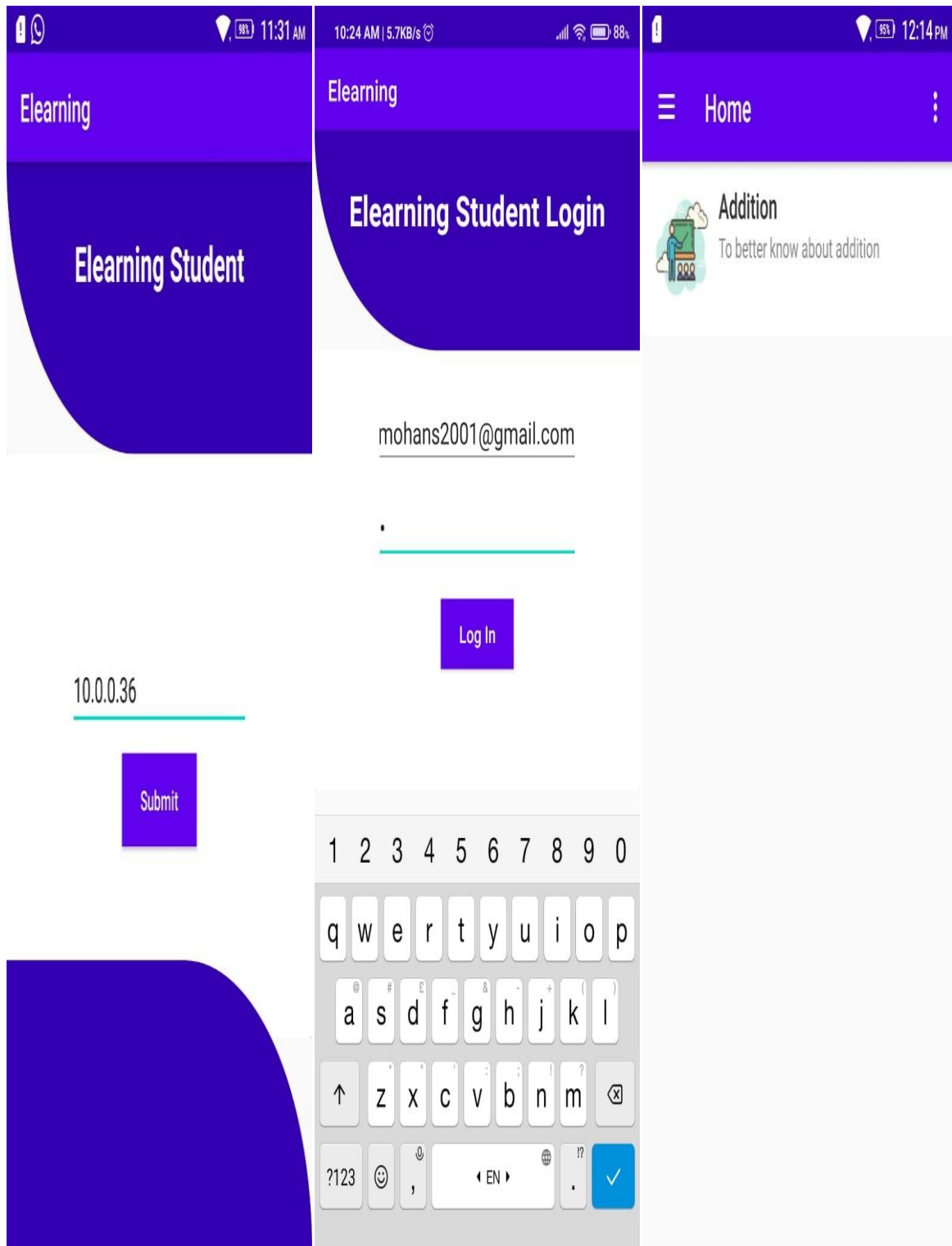
- Status: **Course Started** (green text)
- Action: **Delete Course** (button)
- Image: Two women looking at a laptop.
- Subject: **Science**
- Topic: Life Organism
- Activities: **1 Activities** (with **View Activities** button)
- Status: **Course Started** (green text)
- Action: **Delete Course** (button)

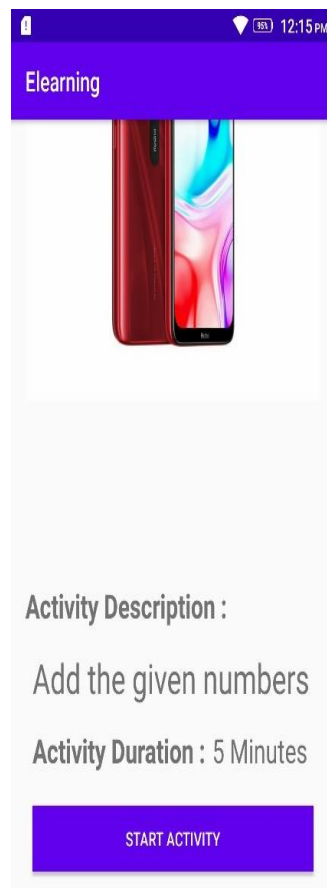
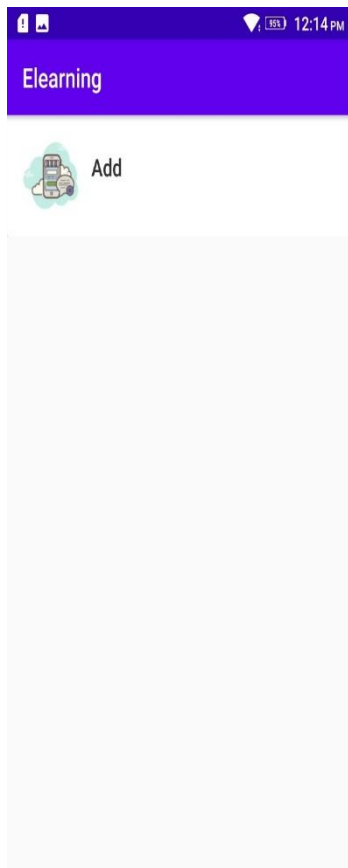
**Course 2 (Right Column):**

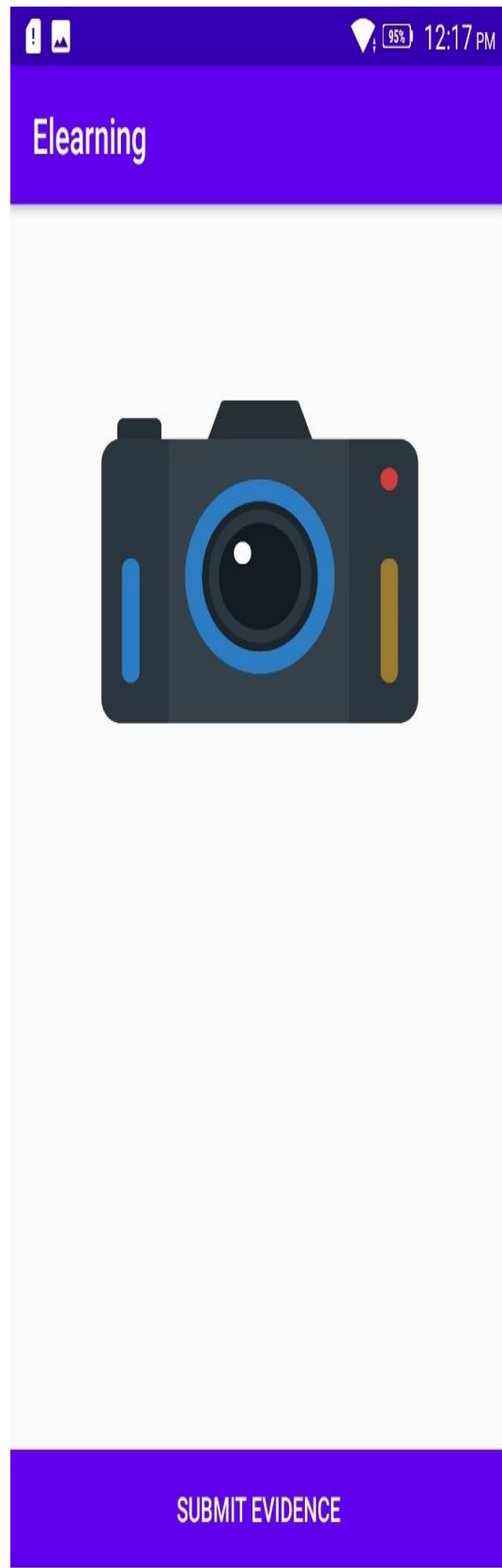
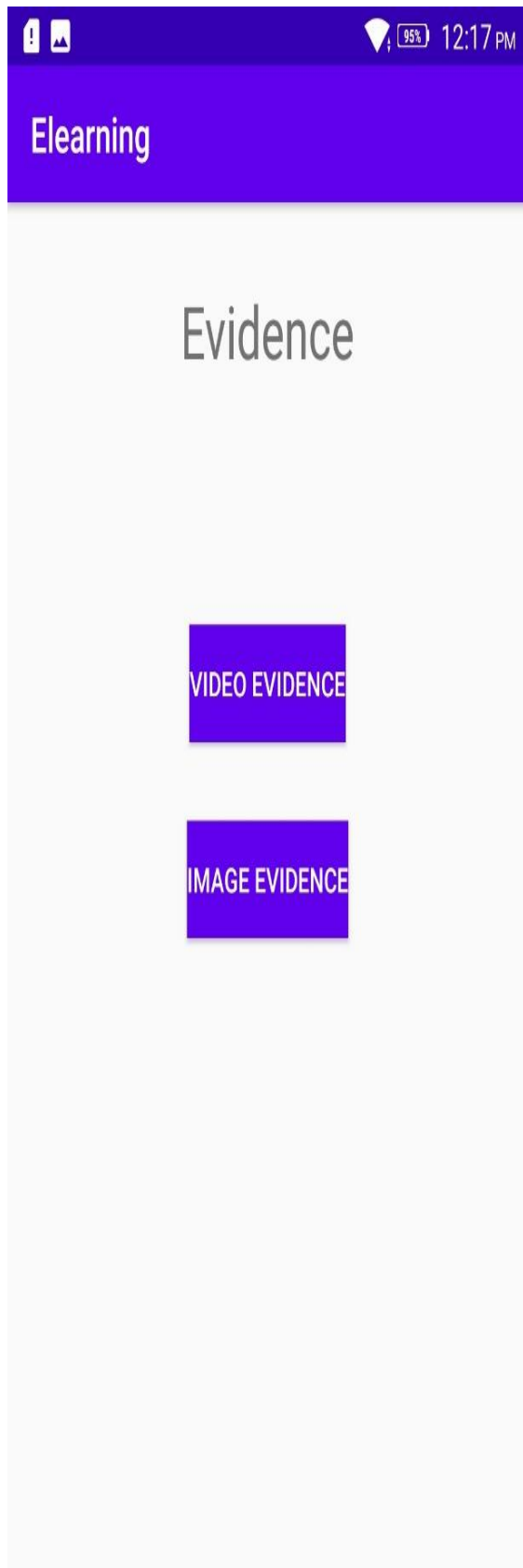
- Status: **Course Started** (green text)
- Action: **Delete Course** (button)
- Image: Two women looking at a laptop.
- Subject: **Tamil**
- Topic: grammar
- Activities: **1 Activities** (with **View Activities** button)
- Status: **Course Started** (green text)
- Action: **Delete Course** (button)

The Windows taskbar at the bottom shows the search bar, task view button, and several application icons. The system clock indicates 10:22 on 11-03-2022.

## SMART PHONE TEST ACTIVITY







***SELECTED IMAGE WHERE FACE DETECTION BY THE MALEMO WAS  
COMPUTED***



((1)) ((2)) ((3)) ((4)) ((5)) ((6)) ((7))



((8)) ((9)) ((10)) ((11)) ((12)) ((13)) ((14))



((15)) ((16)) ((17)) ((18)) ((19)) ((20)) ((21))

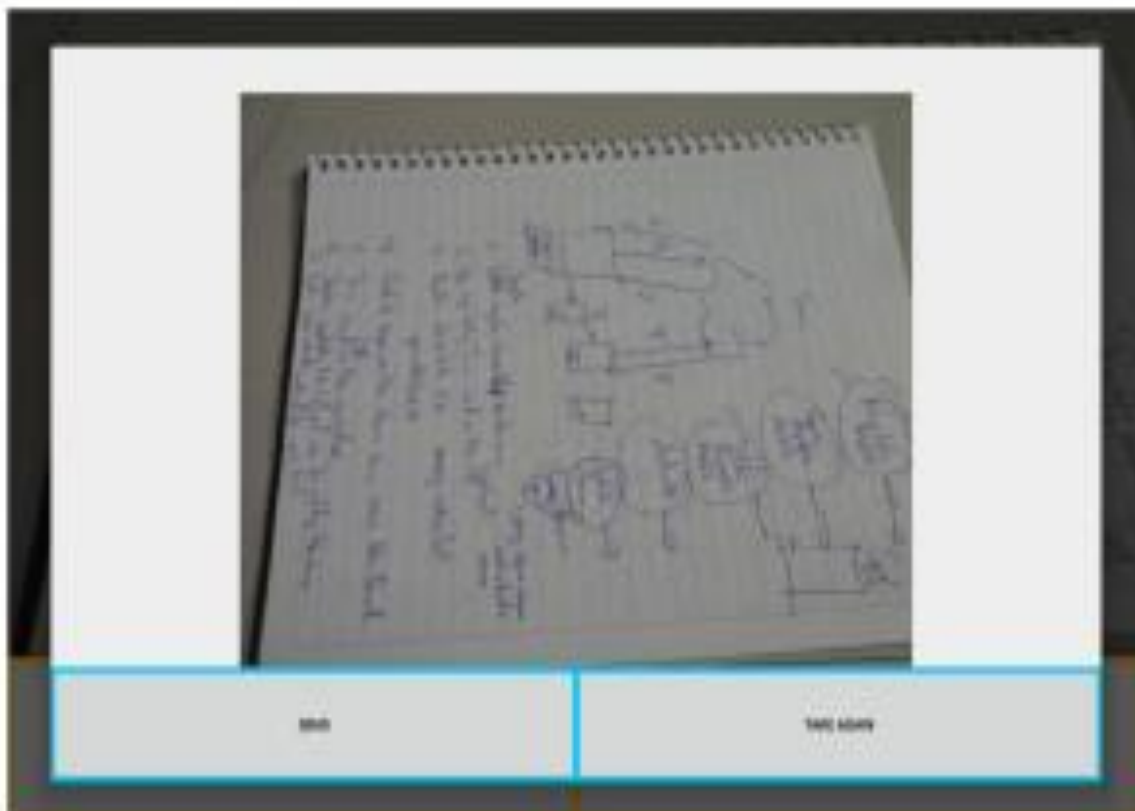
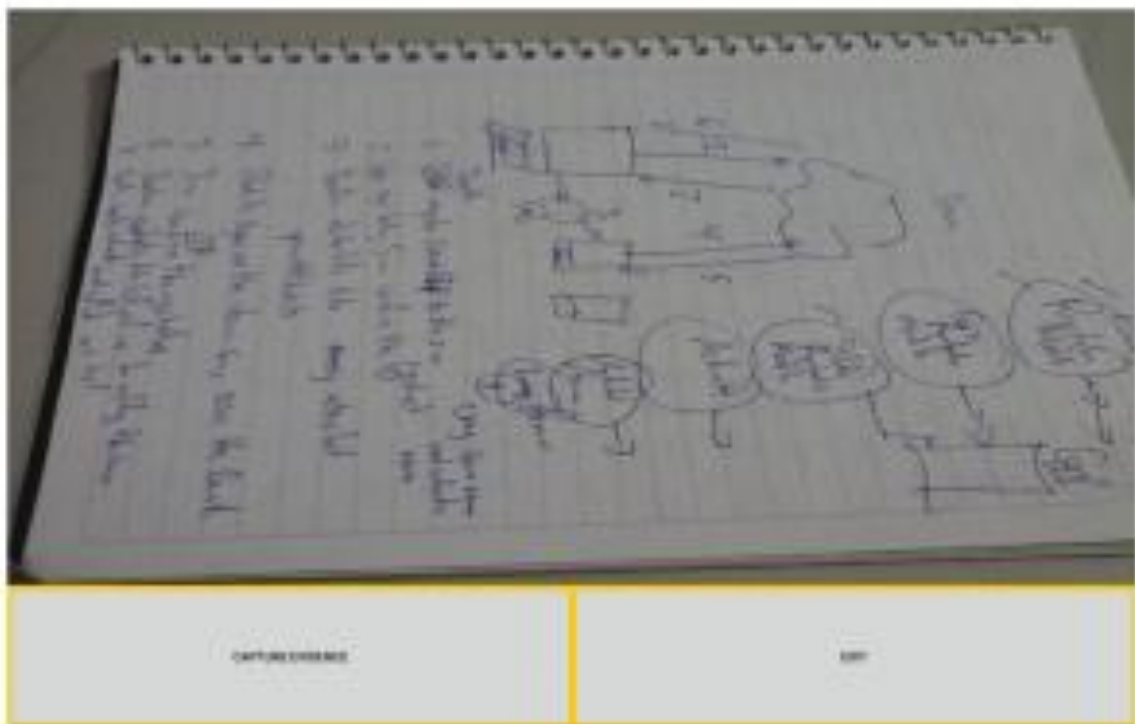


((22)) ((23)) ((24)) ((25)) ((26)) ((27)) ((28))

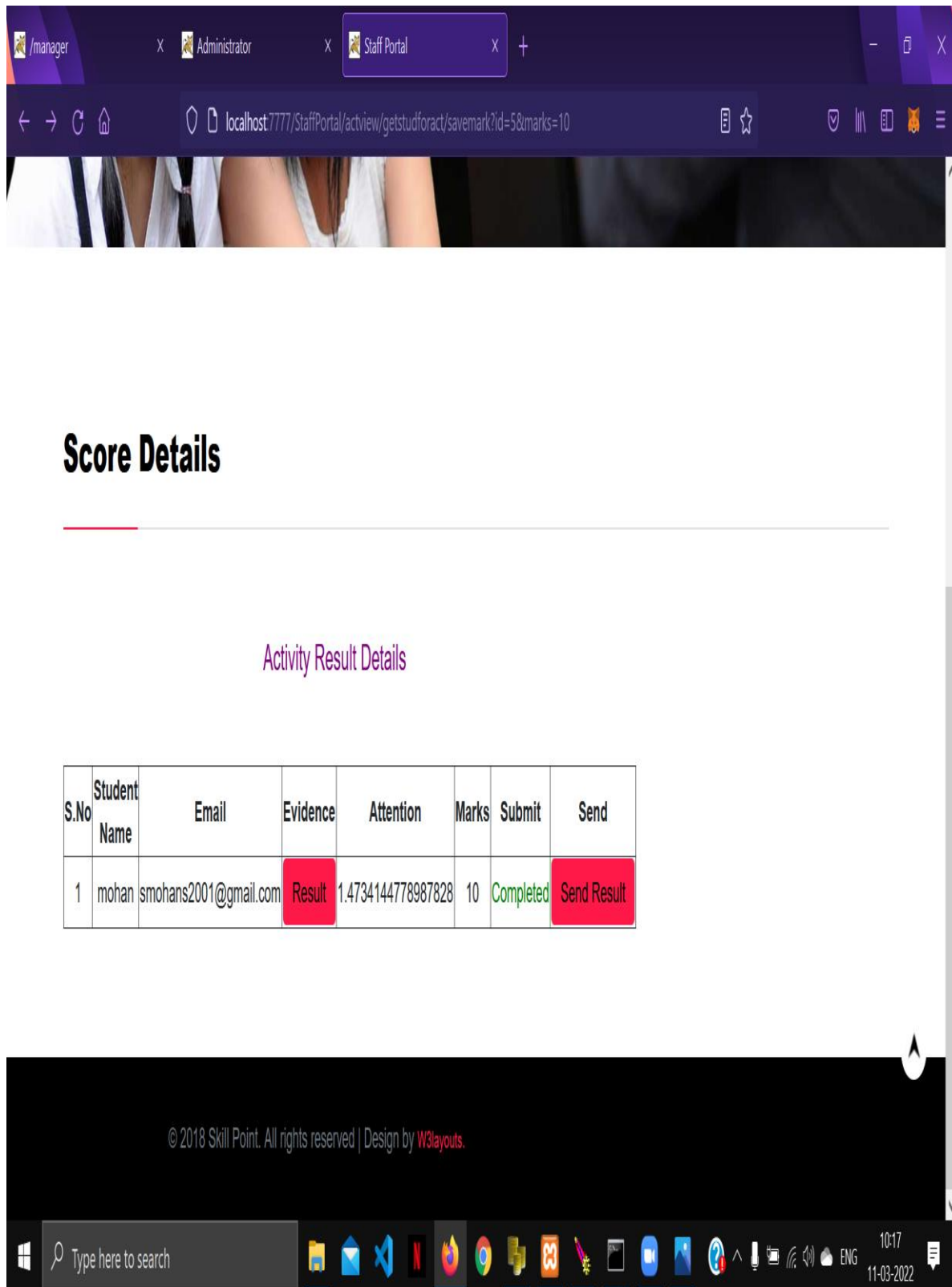


((29)) ((30)) ((31)) ((32)) ((33)) ((34))

## EVIDENCE CAPTURING SCREEN



## SCORE DETAILS



Score Details

Activity Result Details

| S.No | Student Name | Email                 | Evidence | Attention          | Marks | Submit    | Send        |
|------|--------------|-----------------------|----------|--------------------|-------|-----------|-------------|
| 1    | mohan        | smohans2001@gmail.com | Result   | 1.4734144778987828 | 10    | Completed | Send Result |

© 2018 Skill Point. All rights reserved | Design by W3layouts.

## REFERENCES

[1] J. Steil, P. Müller, Y. Sugano, and A. Bulling, "Forecasting user attention during everyday mobile interactions using device-integrated and wearable sensors," in Proc. 20th Int. Conf. Hum.-Comput. Interact. Mobile Devices Services. New York, NY, USA: Association for Computing Machinery, Sep. 2018, pp. 1\_13.

[2] J. Kangas, D. Akkil, J. Rantala, P. Isokoski, P. Majaranta, and R. Raisamo, "Gaze gestures and haptic feedback in mobile devices," in Proc. 32nd Annu. ACM Conf. Hum. Factors Comput. Syst. (CHI). New York, NY, USA: Association for Computing Machinery, 2014, pp. 435\_438.

[3] M. Tonsen, J. Steil, Y. Sugano, and A. Bulling, "Invisibleeye: Mobile eye tracking using multiple low-resolution cameras and learning-based gaze estimation," in Proc. ACMInteract. MobileWearable Ubiquitous Technol.,

vol. 1, Sep. 2017, pp. 1\_21.

[4] C. Mesquita and R. P. Lopes, "Head posture detection for measuring concentration," in Proc. INNODOCT/13 New Changes Technol. Innov., 2013, pp. 1\_9.

[5] I. Teyeb, O. Jemai, M. Zaided, and C. Ben Amar, "A novel approach for drowsy driver detection using head posture estimation and eyes recognition based on wavelet network," in Proc. 5th Int. Conf. Inf., Intell., Syst. Appl. (IISA), Jul. 2014, pp. 379\_384.

[6] A. Hutanu and P. E. Berteau, "A review of eye tracking in elearning," in Proc. 15th Int. Sci. Conf. eLearn. Softw. Educ., 2019, pp. 281\_287.

[7] E. Chong, N. Ruiz, Y. Wang, Y. Zhang, A. Rozga, and J. M. Rehg, "Connecting gaze, scene, and attention: Generalized attention estimation via joint modeling of

gaze and scene saliency," in Proc. 15th Eur. Conf. Comput. Vis. (ECCV), in Lecture Notes in Computer Science, vol. 11209, V. Ferrari, M. Hebert, C.

Sminchisescu, and Y. Weiss, Eds. Munich, Germany: Springer, Sep. 2018, pp. 397\_412.

[8] L. Copeland, "Eye tracking to support eLearning," M.S. thesis, Res. School Comput. Sci., College Eng. Comput. Sci., Austral. Nat. Univ., Canberra, ACT, Australia, 2016.

[9] J. Waleed, T. M. Hasan, and Q. K. Abed, "Eye-gaze estimation systems for multi-applications: An implementation of approach based on laptop webcam," Diyala J. Pure Sci., vol. 14, no. 2, pp. 153\_173, Apr. 2018.

[10] P. Ricciardelli and J. Driver, "Effects of head orientation on gaze perception: How positive congruency effects can be reversed," Quart. J. Experim. Psychol., vol. 61, no. 3, pp. 491\_504, Mar. 2008.

[11] A. Fathi, Y. Li, and J. M. Rehg, "Learning to recognize daily actions using gaze," in Proc. 12th Eur. Conf. Comput. Vis. Berlin, Germany: Springer-Verlag, 2012, pp. 314\_327.

[12] Z. Jiang, J. Han, C. Qian, W. Xi, K. Zhao, H. Ding, S. Tang, J. Zhao, and P. Yang, "VADS: Visual attention detection with a smartphone," in Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM), Apr. 2016, pp. 1\_9.

[13] D. Xiao and C. Feng, "Detection of drivers visual attention using smartphone," in Proc. 12th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery (ICNC-FSKD), Aug. 2016, pp. 630\_635.

[14] Z. Jiang, K. Zhao, R. Li, and J. Zhao, "I-VALS: Visual attention localization for mobile service computing," *IEEE Access*, vol. 7, pp. 45166\_45181, 2019.

[15] X. Xu and X. Teng, "Classroom attention analysis based on multiple euler angles constraint and head pose estimation," in *MultiMedia Modeling*, Y. M. Ro, W.-H. Cheng, J. Kim, W.-T. Chu, P. Cui, J.-W. Choi, M.-C. Hu, and W. De Neve, Eds. Cham, Switzerland: Springer, 2020, pp. 329\_340.

[16] S. Asteriadis, K. Karpouzis, and S. Kollias, "The importance of eye gaze and head pose to estimating levels of attention," in *Proc. 3rd Int. Conf. Games Virtual Worlds Serious Appl.*, May 2011, pp. 186\_191.

[17] F. Sana, T. Weston, and N. J. Cepeda, "Laptop multitasking hinders classroom learning for both users and nearby peers," *Comput. Educ.*, vol. 62, pp. 24\_31, Mar. 2013.

[18] H. Støle, A. Mangen, and K. Schwippert, "Assessing children's reading comprehension on paper and screen: A mode-effect study," *Comput. Educ.*, vol. 151, Jul. 2020, Art. no. 103861.

[19] A. Chern, Y.-H. Lai, Y.-P. Chang, Y. Tsao, R. Y. Chang, and H.-W. Chang, "A smartphone-based multi-functional hearing assistive system to facilitate speech recognition in the classroom," *IEEE Access* H. L. Egger, G. Dawson, J. Hashemi, K. L. H. Carpenter, S. Espinosa, K. Campbell, S. Brotkin, J. Schaich-Borg, Q. Qiu, M. Tepper, J. P. Baker, R. A. Bloomfeld, and G. Sapiro, "Automatic emotion and

attention analysis of young children at home: A ResearchKit autism feasibility study," NPJ Digit. Med., vol. 1, no. 1, p. 20, Dec. 2018.

[20] P. Smutny and P. Schreiberova, "Chatbots for learning: A review of educational chatbots for the Facebook messenger," Comput. Educ., vol. 151, Jul. 2020, Art. no. 103862.

[21] O. A. H. Jones, M. Spichkova, and M. J. S. Spencer, "Chirality-2: Development of a multilevel mobile gaming app to support the teaching of introductory undergraduate-level organic chemistry," J. Chem. Educ., vol. 95, no. 7, pp. 1216\_1220, Jul. 2018.

[22] D. Mak and D. Nathan-Roberts, "Design considerations for educational mobile apps for young children," in Proc. Hum. Factors Ergonom. Soc. Annu. Meeting, 2017, vol. 61, no. 1, pp. 1156\_1160.

[23] C.-S. Wang, Y.-M. Huang, and K.-S. Hsu, "Developing a mobile game to support students in learning color mixing in design education," Adv. Mech. Eng., vol. 9, no. 2, 2017, Art. no. 1687814016685226.

[24] M. Rainoldi, B. Neuhofer, and M. Jooss, "Mobile eyetracking of museum learning experiences," in Information and Communication Technologies in Tourism, B. Stangl and J. Pesonen, Eds. Cham, Switzerland: Springer, 2018, pp. 473\_485.

[25] P. Lavin-Mera, J. Torrente, P. Moreno-Ger, J. A. V. Pinto, and B. Fernández-Manjón, "Mobile game development for multiple devices in education," Int. J. Emerg. Technol. Learn., vol. 4, no. s2, pp. 1\_8, Oct. 2009.

[26] A. Murphy, H. Farley, M. Lane, A. Hafeez-Baig, and B. Carter, "Mobile learning anytime, anywhere: What are our students doing?" *Australas. J. Inf. Syst.*, vol. 18, no. 3, pp. 1\_15, Nov. 2014.

[27] A. Suner, Y. Yilmaz, and B. Pişkin, "Mobile learning in dentistry: Usage habits, attitudes and perceptions of undergraduate students," *PeerJ*, vol. 7, p. e7391, Jul. 2019.

[28] M. Elkhateeb, A. Shehab, and H. El-Bakry, "Mobile learning system for Egyptian higher education using agile-based approach," *Educ. Res. Int.*, vol. 2019, pp. 1\_13, Mar. 2019.

[29] A. Hendrayana and W. Sanusi, "Mobile learning to improve mathematics teachers mathematical competencies," *J. Phys., Conf. Ser.*, vol. 948, Jan. 2018, Art. no. 012049.

[30] G. Silva-Maceda, P. D. Arjona-Villicana, and F. E. Castillo-Barrera, "More time or better tools? A large-scale retrospective comparison of pedagogical approaches to teach programming," *IEEE Trans. Educ.*, vol. 59, no. 4, pp. 274\_281, Nov. 2016.