

Name of the course: Android App Development

Topic: Deadlock Model

Course Incharge: Dr. S. Kavitha

Innovative Method: Flipped Classroom



The classroom session focused on application-based learning rather than theoretical explanation. The following activities were conducted:

Recap and Clarification

The session began with a quick discussion to address doubts from the pre-class materials, ensuring that all students had a clear understanding of the basics.

Group Activity – Code Analysis

Students were divided into small groups, and each group received an **Android code snippet** containing a potential deadlock situation (e.g., nested synchronized blocks or locking UI thread).

The task was to:

- Identify the point of deadlock
- Explain how and why it occurs
- Predict the behaviour (e.g., ANR)
- Suggest solutions

Hands-On Demonstration

A live demonstration showing a simulated **Application Not Responding (ANR)** scenario caused by deadlock was presented. Students observed how thread blocking impacts app performance.

Solution Discussion

Groups presented their findings, and the faculty facilitated a discussion on:

- Avoiding nested locks
- Using timeout-based locking mechanisms
- Designing thread-safe components
- Best practices for Android concurrency

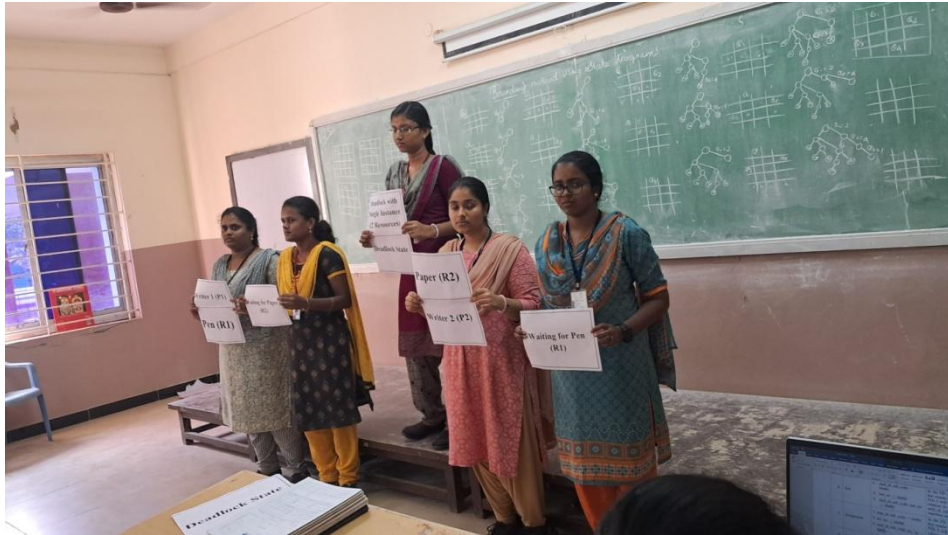
Velammal College of Engineering and Technology, Madurai - 625 009
(Autonomous)
Department of Computer Science and Engineering (Cyber Security)
Innovative Teaching Methodology

Name of the course: Software Testing

Topic: Levels of Testing

Course Incharge: Mr. S. Murali

Innovative Method: Role Play



A **Role Play activity** was conducted for the topic *Levels of Testing* in Unit III of the Software Testing course to enhance student engagement and conceptual clarity. Students were assigned label sheets representing different testing levels such as Unit Testing, Integration Testing, System Testing, Acceptance Testing, Regression Testing, Alpha Testing and Beta Testing.

Each student enacted their role, demonstrating how software moves through various stages of testing before release. Through this interactive activity, students visualized the flow of testing, identified the purpose of each level, and understood how defects are detected and resolved during the testing cycle. The enactment helped them relate theoretical concepts to real-world software development practices. The activity encouraged teamwork, communication and active participation, making the learning process more lively and memorable. Students expressed that the role play helped them retain concepts better compared to a traditional lecture. Overall, the methodology proved effective in building a strong understanding of testing stages and improving students' analytical and collaborative skills.