Graphic Era Hill University
DEHRADUN • BHIMTAL • HALDWANI

PROJECT AND TEAM INFORMATION

Project Title

Deadlock Simulator using Python

Student/Team Information

Team Name	MEGARUSHERS
Team Member 1 (Team Lead) Samarth Agarwal Student ID: - 22011896 samarth2404agarwal@gmail.com	
Team Member 2 Kunwardeep Singh Student ID: - 22011787 kunwar2104@gmail.com	

Team Member 3 Lakshaydeep Chaudhary Student ID: - 2219016 lakshay71003@gmail.com	

PROJECT PROGRESS DESCRIPTION

Project Abstract

The Deadlock Simulator is a Python-based tool designed to simulate, visualize, and analyze deadlock scenarios in operating systems. It demonstrates the occurrence of deadlocks via interactive scenes such as Exam Hall and Library, highlighting conditions like Mutual Exclusion, Hold and Wait, No Preemption, and Circular Wait. The simulator enables understanding of detection and prevention mechanisms, including Banker's Algorithm, through process-resource simulations. The goal is to make deadlock handling more practical and understandable for learners.

Updated Project Approach and Architecture

The simulator uses a modular architecture:

1. Core Simulation – Implements resources, processes, and deadlock logic using Python's threading, lock, and collections.
2. Deadlock Detection – Resource Allocation Graph (RAG) applied in Exam Hall simulation.
3. Deadlock Avoidance/Prevention – Priority Scheduling (FIFO) in Library simulation; Banker's Algorithm integration completed.
4. Visualization – GUI fully developed using Tkinter.
5. User Interface – GUI and CLI implemented.

Technologies Used: Python 3.x, threading, time, sys, collections, matplotlib, networkx, Tkinter.

Tasks Completed

Task Completed	Team Member
Exam Hall Deadlock Simulation	Samarth Agarwal
Library Priority Scheduling	Kunwardeep Singh
Optimized Library Scene	Lakshaydeep Chaudhary
Banker's Algorithm Integration	Kunwardeep Singh
GUI Development (Tkinter)	Lakshaydeep Chaudhary
Comprehensive Testing & Debugging	Samarth Agarwal

Challenges/Roadblocks

All earlier roadblocks such as synchronization, input validation, and visualization limitations have been resolved. GUI and Banker's Algorithm integration are now successfully completed.

Tasks Pending

None. All project tasks are completed.

Project Outcome/Deliverables

Fully Functional Python-based Deadlock Simulator
CLI and GUI both implemented
Deadlock Detection (Exam Hall) working
Library FIFO Scheduling working
Banker's Algorithm fully integrated
Comprehensive Testing Done
Documentation completed

Progress Overview

100% Complete
No delays, no pending modules—Project completed as per plan.

Codebase Information

Repository:
https://github.com/MegarusherSamarth/PBL/tree/main/Software%20Engineering%20(OS)

Testing and Validation Status

Test Type	Status	Notes
Exam Hall Deadlock Test	Pass	Deadlock condition detection verified.
Library FIFO Scheduling	Pass	FIFO order for Teacher/Student verified.
Banker's Algorithm Test	Pass	Algorithm produces safe sequence correctly.

Deliverables Progress

Deliverable	Status
Exam Hall Deadlock Simulation	Completed
Library FIFO Priority Scheduling	Completed
Banker's Algorithm Implementation	Completed
GUI Development	Completed
Final Documentation & Report	Completed