

Final Project

Classroom Management System

Course Code: CPE 201L	Program: BSCPE
Course Title: DATA STRUCTURES AND ALGORITHMS	Date Performed: October 11, 2025
Section: CpE 2-A	Date Submitted: November 8, 2025
Leader: Filjohn Delinia	
Members: Czer Justine Maringal Paul Justine Polestico Mark Angel Talagtag	Instructor: Engr. Maria Rizette Sayo

1. Objective(s):

1. Primary Objective: To demonstrate the implementation and application of FIFO (First-In-First-Out) queue data structure in a real-world classroom booking system
2. Functional Objectives:
 - Manage classroom and laboratory booking requests efficiently
 - Prevent scheduling conflicts for room reservations
 - Automatically handle expired bookings
 - Provide real-time availability checking
 - Maintain persistent storage using Excel files

2. Intended Learning Outcomes (ILOs):

Upon studying this system, users should be able to:

1. Understand the practical implementation of FIFO queue operations (enqueue, dequeue, peek)
2. Analyze how data structures solve real-world scheduling problems
3. Implement conflict detection algorithms for resource allocation
4. Design automated expiry systems for time-based reservations
5. Develop GUI applications with persistent data storage

3. Discussion:

FIFO Queue Implementation

The system implements a true FIFO queue where:

- Enqueue: New booking requests are added to the end of the queue
- Dequeue: The oldest request (front of queue) is processed first
- Peek: View the next request without removing it

KEY FEATURES

1. Conflict Detection: Prevents double-booking by checking time overlaps
2. Smart Suggestions: Recommends next available time slots when conflicts occur
3. Automatic Expiry: Removes past bookings using time-based validation
4. Persistent Storage: Excel integration for data preservation
5. Operation Logging: Tracks all queue operations for demonstration purposes

TECHNICAL ARCHITECTURE

- Frontend: Tkinter GUI with intuitive form inputs
- Backend: Custom FIFO queue with booking validation
- Storage: Pandas Excel integration
- Scheduling: Automated expiry checking using system timers

4. Materials and Equipments:

Software Requirements

- Python 3.7+
- Required libraries:
 - tkinter (GUI framework)
 - pandas (Excel file handling)
 - tkcalendar (date selection widget)
 - datetime (time manipulation)
 - logging (system monitoring)

Hardware Requirements

- Standard computer system
- Minimum 4GB RAM
- 500MB storage space

SYSTEM COMPONENTS

1. BookingRequest Class: Data structure for reservation details
2. RoomBookingQueue Class: FIFO queue implementation
3. TimeUtils Class: Time conversion and validation utilities
4. ExcelManager Class: Data persistence handler
5. BookingApp Class: Main application controller

5. Procedure:

SYSTEM OPERATION FLOW

1. Initialization:
 - Load existing bookings from Excel file
 - Initialize FIFO queue with historical data
 - Schedule first expiry check
2. Booking Submission:
 - User fills booking form (name, room type, date, time)
 - System validates input and checks for conflicts
 - If valid, request is enqueued (added to queue end)
 - Excel file is updated automatically
3. Queue Management:
 - Process Next: Dequeue oldest request (FIFO principle)
 - View Next: Peek at front request without removal
 - Delete Selected: Remove specific request by index
 - Automatic Expiry: System periodically removes past bookings
4. Conflict Resolution:
 - Time overlap detection for same room/date
 - Smart suggestion of next available slots
 - Visual feedback to user

FIFO DEMONSTRATION STEPS

1. Submit multiple booking requests in sequence
2. Observe queue order in "Pending Requests" list
3. Use "Process Next Request" to demonstrate FIFO dequeue
4. Check operation log to track queue modifications
5. Note how oldest requests are always processed first

6. Output

Visual Outputs

GUI Interface:

- Booking form with dropdown selections
- Real-time queue display in FIFO order
- Operation buttons for queue management
- Status indicators

System Messages:

- Success/error notifications
- Conflict warnings with suggestions
- FIFO operation confirmations
- Expiry removal alerts

Data Outputs:

- Excel file (booking_requests.xlsx) with all active bookings
- FIFO operation log showing queue modifications
- System logs for debugging and monitoring

Queue Behavior Output

- Requests maintain insertion order
- Processing always follows chronological order
- Visual queue position indicators ([1], [2], [3], etc.)
- Real-time queue size tracking

Classroom Management System - FIFO Demonstration

Booking Details

Year/Section:

Room Type: Room Number / Lab Name:

Date: 11/8/25

Start Time: 9 00 AM

Subject Units: 1 Unit 2 Units 3 Units 4 Units

Lecture: 1 unit = 1 hour | Laboratory: 1 unit = 3 hours

Pending Requests (FIFO Order)

FIFO Operations Demo

Queue: 0 request(s)

7. Conclusion:

1. FIFO Efficiency: The queue structure efficiently manages booking requests in chronological order, ensuring fair first-come-first-served processing
2. Conflict Prevention: The system successfully prevents double-booking through comprehensive time overlap detection
3. Automation Benefits: Automatic expiry handling reduces manual maintenance and keeps the queue current
4. User Experience: The GUI provides intuitive access to complex queue operations

Educational Value

This implementation demonstrates:

- Practical application of fundamental data structures
- Real-world problem-solving with algorithms
- GUI integration with backend logic
- Persistent data management techniques
- Automated system maintenance

Potential Enhancements

1. Priority queue implementation for emergency bookings
2. Multi-room simultaneous booking capability
3. Advanced conflict resolution algorithms
4. Web-based interface for remote access
5. Real-time notifications system

8. References

[1] Python Software Foundation, “Python 3.11 Documentation,” Python.org, 2023. [Online].

<https://docs.python.org/3.11/>

[2] Pandas Development Team, “Pandas User Guide,” pandas.pydata.org, 2023. [Online].

<https://pandas.pydata.org/docs/>

[3] “GUI Programming with Tkinter,” Tkinter Documentation, 2023. [Online].

<https://docs.python.org/3/library/tkinter.html>

[4] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., “Introduction to Algorithms – Queue Data Structures,” MIT Press, 2009. [Online].

<https://mitpress.mit.edu/9780262033848/introduction-to-algorithms/>

[5] Knuth, D. E., “The Art of Computer Programming – Fundamental Algorithms,” Addison-Wesley, 1997. [Online].

<https://www-cs-faculty.stanford.edu/~knuth/taocp.html>

[6] “Data Structures and Algorithms Curricula,” Association for Computing Machinery (ACM), 2023. [Online].

<https://www.acm.org/education/curricula>

[7] “Queue Theory and Applications,” Tutorials Point, 2023. [Online].

https://www.tutorialspoint.com/queue_data_structure/index.htm

[8] “Resource Scheduling Systems Design,” GeeksforGeeks, 2023. [Online].

<https://www.geeksforgeeks.org/resource-allocation-and-scheduling/>