DS2030 Data Structures and Algorithms for Data Science
Lab 2

August 19th, 2025

Lab Instructions

- Create a folder named **"DS2030_<RollNo.>"** (all letters in capital) in **"home"** directory. Eg- **DS2030_142402022**
- Load the given two files into the created folder (we will provide two files to you):
  - A `attendance_system.py` file containing the starter code (you must write your code in this file).
  - A `test file` you have run this to check test results.
- Ensure that all classes, functions, and attributes are named exactly as specified in `attendance_system.py`
- Make sure to save your progress before leaving the lab.
- Do not shut down the system after completing the exam.
- You are not allowed to share code with your classmates nor allowed to use code from the internet. If we find that you have copied code from your classmate or from the Internet, you will get a straight fail grade in the course.

1 Employee Attendance Management using Doubly Circular Linked List

The Employee Attendance Management System is implemented using a **Doubly Circular Linked List (DCLL)**. Each employee is represented as a node in the list, uniquely identified by an `Employee ID`. The system supports attendance operations (check-in/check-out) along with basic CRUD functionalities.

1.1 Employee Class

The `Employee` class encapsulates the following attributes:

- `id`: Unique identifier for the employee
- `name`: Name of the employee
- `check_in_time`: Time when employee checked in
- `check_out_time`: Time when employee checked out
- `is_present`: Boolean flag for attendance
- `next`: Pointer to the next employee in the list
- `prev`: Pointer to the previous employee in the list

2 AttendanceSystem Class

The `AttendanceSystem` class manages all employees using a Doubly Circular Linked List (DCLL). It maintains:

- `head`: Pointer to the first employee in the list
- `total_employees`: Total number of employees in the system

2.1 get_current_time(self)

Implement a function to return the current system time.

- Returns the current system time as a formatted string.

2.2 add_employee(self, emp_id, name)

Implement a function to add a new employee to the system.

- `emp_id`: Unique identifier for the employee.
- `name`: Name of the employee.
- If the list is empty, `head` points to the new employee.
- Otherwise, add the employee to the end of the circular doubly linked list.
- Print a confirmation message after successful addition.

2.3 search_employee(self, emp_id)

Implement a function to search for an employee by ID.

- `emp_id`: Employee ID to search.
- Traverses the list to find the employee.
- Returns the `Employee` object if found, else `None`.

2.4 check_in(self, emp_id)

Implement a function to mark an employee as present and record their check-in time.

- `emp_id`: Employee ID to check in.
- If employee not found, print: `"Employee with ID {emp_id} not found!"`.
- If employee is already present, print: `"Employee {emp.name} is already checked in!"`.
- Otherwise:
  - Mark the employee as present.
  - Set `check_in_time` to the current system time.
  - Reset `check_out_time`.
  - Print confirmation message: `"Employee {emp.name} checked in at {emp.check_in_time}"`.

2.5 check_out(self, emp_id)

Implement a function to mark an employee as absent and record their check-out time.

- `emp_id`: Employee ID to check out.
- If employee not found, print: `"Employee with ID {emp_id} not found!"`.
- If employee is already absent, print: `"Employee {emp.name} is not checked in!"`.
- Otherwise:
  - Mark the employee as absent.
  - Set `check_out_time` to the current system time.
  - Print confirmation message: `"Employee {emp.name} checked out at {emp.check_out_time}"`.

2.6 display_all(self)

Implement a function to display all employees with their attendance status.

- Traverses the entire list.
- Prints each employee's:
  - Employee ID
  - Name
  - Check-in time
  - Check-out time
  - Attendance status

2.7 update_employee(self, emp_id, new_name)

Implement a function to update the name of an employee.

- `emp_id`: Employee ID to update.
- `new_name`: Updated employee name.
- If employee not found, print a message.
- Otherwise, update the name and print a confirmation message.

2.8 delete_employee(self, emp_id)

Implement a function to delete an employee from the system.

- `emp_id`: Employee ID of the employee to be deleted.
- If employee not found, print a message.
- Otherwise:
  - Adjust `next` and `prev` pointers to remove the employee.
  - Update `head` if necessary.
  - Decrement total employees.
  - Print a confirmation message.

2.9 sort_by_id(self)

Implement a function to sort employees in ascending order of their Employee ID.

- Sorts employees based on `emp_id`.
- Updates the circular doubly linked list connections accordingly.
- Print a confirmation message after sorting.

3 Testing

Finally, you have to run the command "./test" in terminal.

Output

```
Employee Alice added successfully!
Employee Bob added successfully!
Employee Charlie added successfully!
Add Employees: PASSED (+1 pts)
Employee Alice checked in at 2025-08-18 19:10:41
Check-in Employee: PASSED (+1 pts)
Employee Alice checked out at 2025-08-18 19:10:41
Check-out Employee: PASSED (+1 pts)
Search Employee: PASSED (+1 pts)
Employee name updated from Bob to Bobby
Update Employee Name: PASSED (+1 pts)
Employee David added successfully!
Employees sorted by ID!
Sort Employees by ID: PASSED (+1 pts)
Employee Charlie deleted successfully!
Delete Employee: PASSED (+1 pts)
Employee Bobby checked in at 2025-08-18 19:10:41
Check-in Employee: PASSED (+0.5 pts)
Employee David is not checked in!
Check-out Employee : PASSED (+0.5 pts)
Employee with ID 105 not found!
Check-in Employee: PASSED (+0.5 pts)
Employee with ID 106 not found!
Delete Employee: PASSED (+0.5 pts)
Employee Bobby is already checked in!
Check-in Employee: PASSED (+0.5 pts)

=== Employee Attendance Status ===
ID    Name                Status     Check-in                  Check-out
--------------------------------------------------------------------------------
100   David               Absent     -                         -
101   Alice               Absent     2025-08-18 19:10:41       2025-08-18 19:10:41
102   Bobby               Present    2025-08-18 19:10:41       -
--------------------------------------------------------------------------------
Display All: PASSED (+0.5 pts)
Passed: 13/13 tests
Total Score: 10.0/10.0 points
```