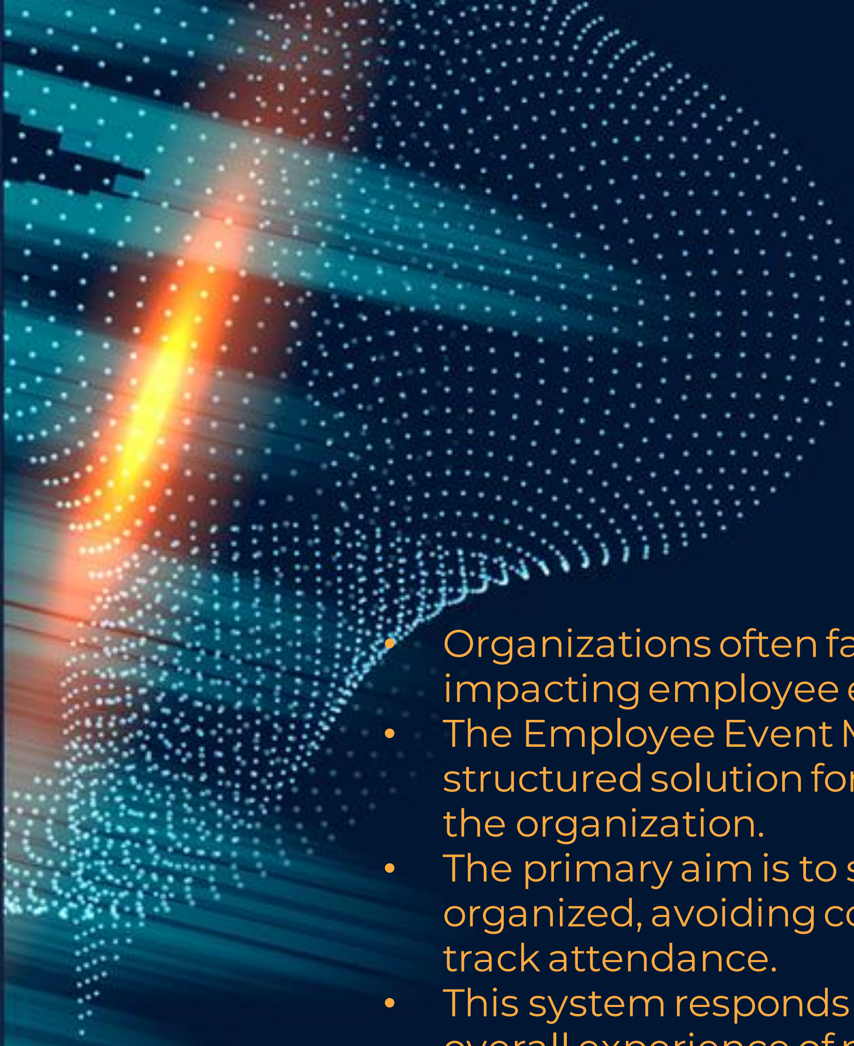
INTERNAL EVENT MANAGEMENT SYSTEM

Presented by:
Ajeet Kumar K Dani
Ismail D
Farhan Jas
Manjunath Swamy



PROBLRM STATEMENT

- Organizations often face challenges in efficiently managing internal events, impacting employee engagement and communication.
- The Employee Event Management System addresses this by providing a structured solution for creating, reading, updating, and deleting events within the organization.
- The primary aim is to streamline the process, ensuring that events are well-organized, avoiding conflicts in scheduling, and providing a mechanism to track attendance.
- This system responds to the need for a centralized tool that enhances the overall experience of planning and executing internal events.

CONTENTS

INTRODUCTION

REQUIREMENT ANALYSIS

FLOWCHART

FUNCTIONS

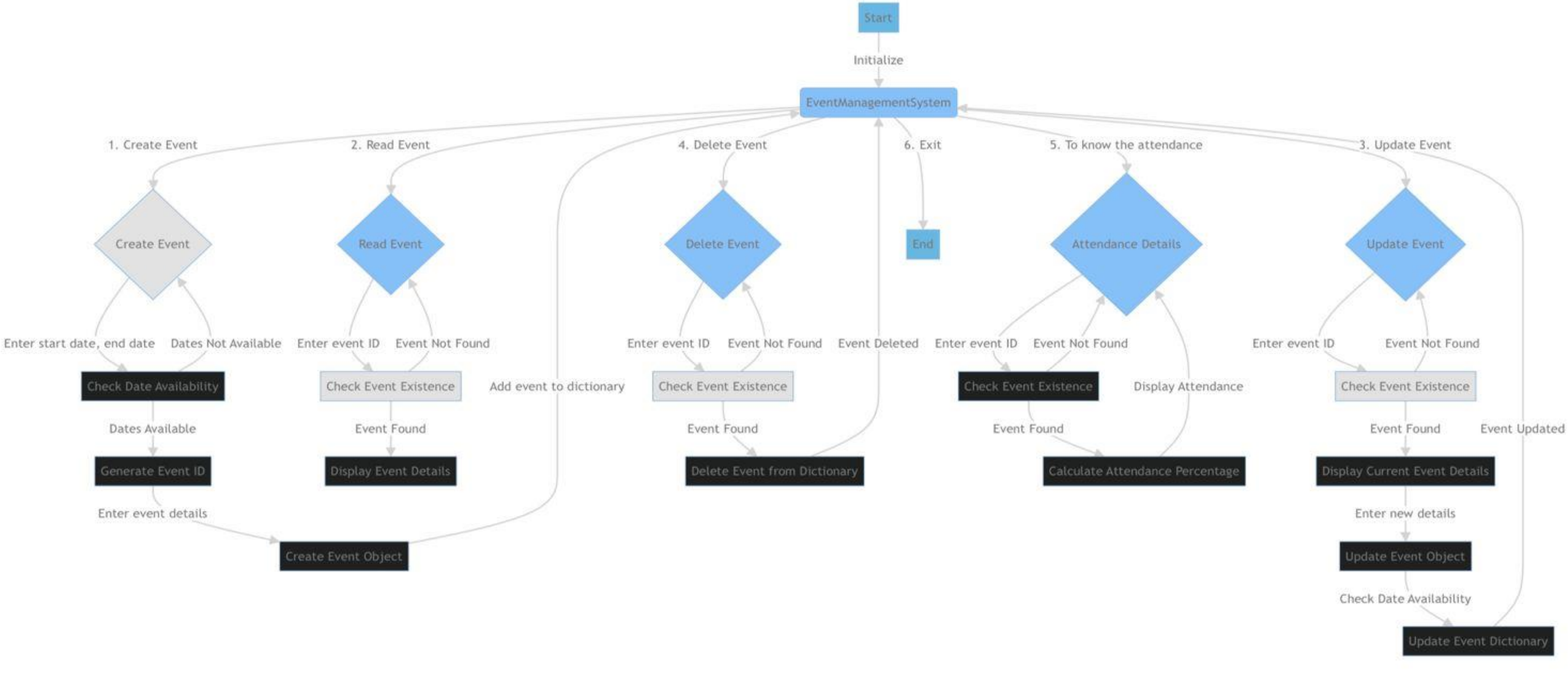
01 INTRODUCTION

- An internal event management system is a vital tool designed to streamline and organize events within an organization.
- Whether it's meetings, conferences, workshops, or team-building activities, this system facilitates the planning, execution, and monitoring of internal events.
- By leveraging technology, it centralizes event-related tasks, ranging from event creation and attendee registration to tracking attendance and updating event details.
- This system often features user-friendly interfaces for event organizers and participants, offering seamless communication and coordination.
- Key functionalities typically include event creation with details such as date, time, and location, attendee management, real-time updates, and post-event analysis.

REQUIREMENT ANALYSIS

- The system's functional requirements focus on providing essential capabilities such as creating, reading, updating, and deleting events.
- A critical functionality is in place to prevent date conflicts when updating events, ensuring the integrity of the event schedule.
- The attendance tracking feature fulfills the need for insights into employee participation, contributing to a more engaged workforce. On the non-functional side, the system is designed to offer a user-friendly command-line interface, making it accessible to users with varying technical backgrounds.
- Security measures are implemented to safeguard data, ensuring that only authorized users can interact with the system.
- These requirements collectively contribute to an effective and secure internal event management system.

FLOWCHART



Start

Initialize

EventManagementSystem

1. Create Event

2. Read Event

4. Delete Event

6. Exit

5. To know the attendance

3. Update Event

Create Event

Read Event

Delete Event

End

Attendance Details

Update Event

Enter start date, end date

Dates Not Available

Enter event ID

Event Not Found

Enter event ID

Event Not Found

Event Deleted

Enter event ID

Event Not Found

Enter event ID

Event Not Found

Check Date Availability

Check Event Existence

Add event to dictionary

Check Event Existence

Check Event Existence

Display Attendance

Check Event Existence

Dates Available

Event Found

Event Found

Event Found

Event Found

Event Updated

Generate Event ID

Display Event Details

Delete Event from Dictionary

Calculate Attendance Percentage

Display Current Event Details

Enter event details

Enter new details

Create Event Object

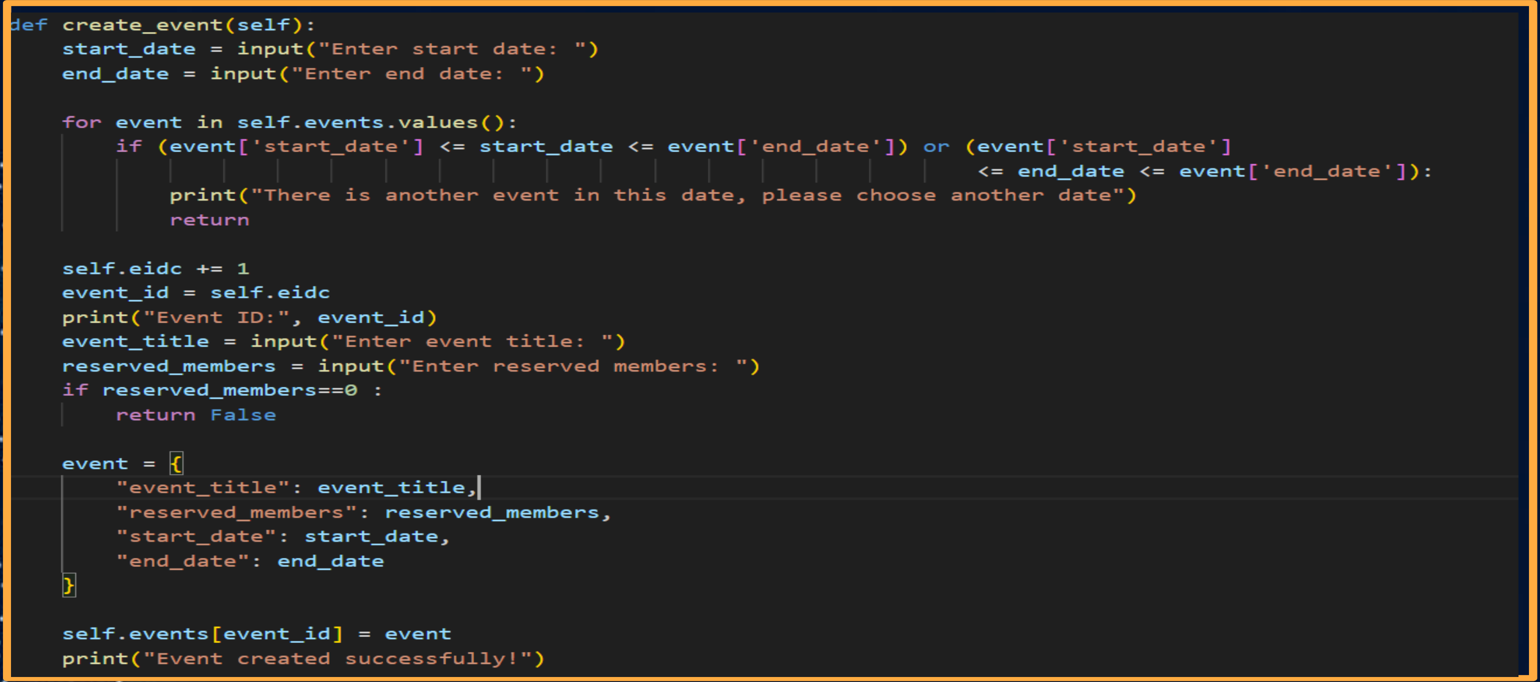
Update Event Object

Check Date Availability

Update Event Dictionary

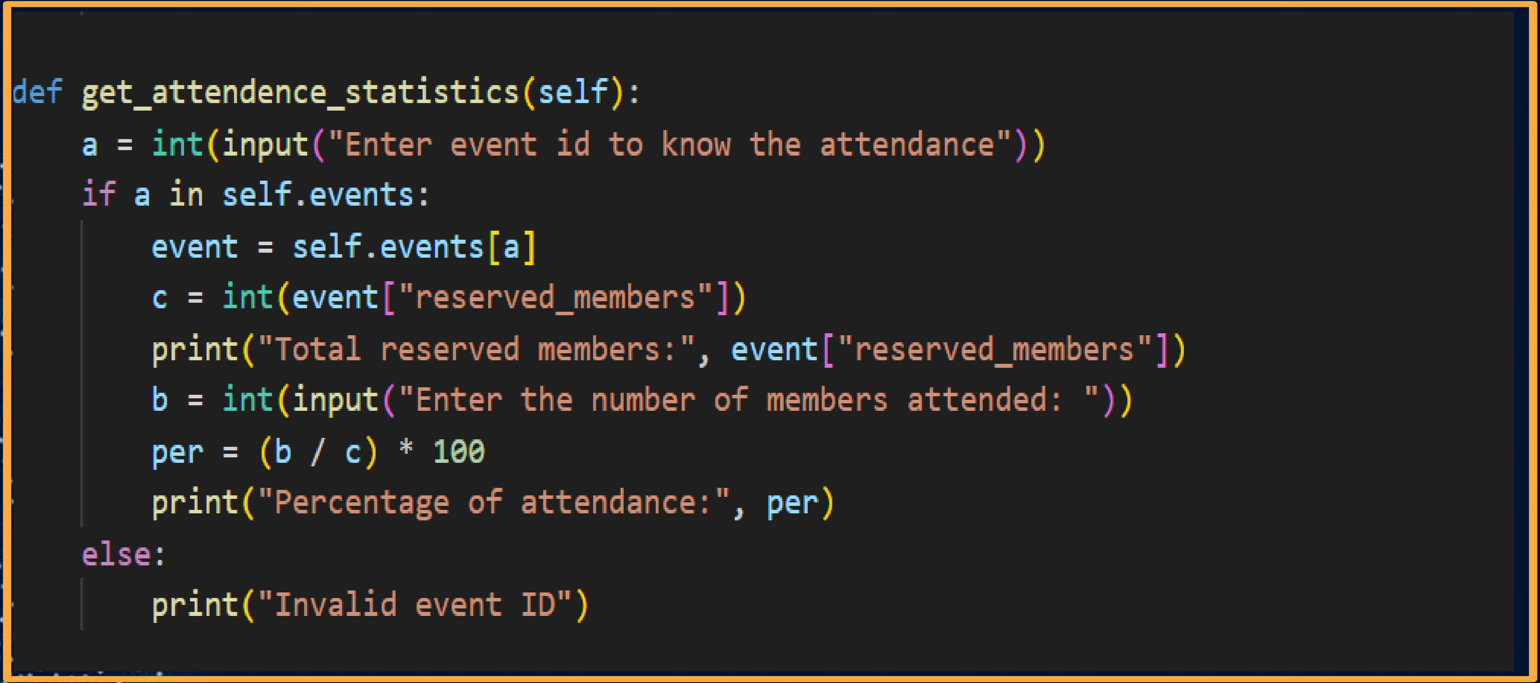
FUNCTIONS

SNIPPET : 1



```python
def create_event(self):
    start_date = input("Enter start date: ")
    end_date = input("Enter end date: ")

    for event in self.events.values():
        if (event['start_date'] <= start_date <= event['end_date']) or (event['start_date']
                                                                        <= end_date <= event['end_date']):
            print("There is another event in this date, please choose another date")
            return

    self.eidc += 1
    event_id = self.eidc
    print("Event ID:", event_id)
    event_title = input("Enter event title: ")
    reserved_members = input("Enter reserved members: ")
    if reserved_members==0 :
        return False

    event = {
        "event_title": event_title,
        "reserved_members": reserved_members,
        "start_date": start_date,
        "end_date": end_date
    }

    self.events[event_id] = event
    print("Event created successfully!")
```

SNIPPET : 2



```python
def get_attendence_statistics(self):
    a = int(input("Enter event id to know the attendance"))
    if a in self.events:
        event = self.events[a]
        c = int(event["reserved_members"])
        print("Total reserved members:", event["reserved_members"])
        b = int(input("Enter the number of members attended: "))
        per = (b / c) * 100
        print("Percentage of attendance:", per)
    else:
        print("Invalid event ID")
```

WHAT ELSE?

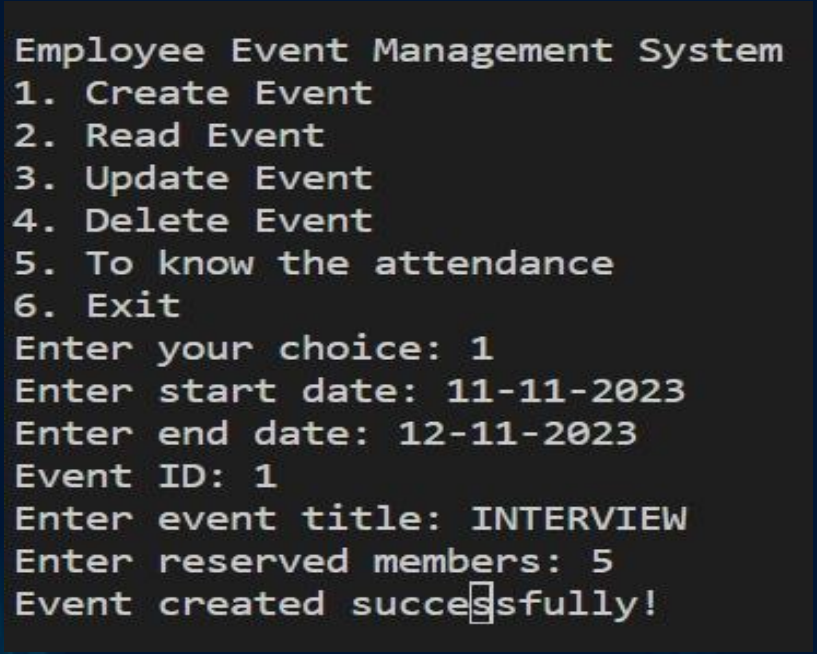
OUTPUT

CONCLUSION

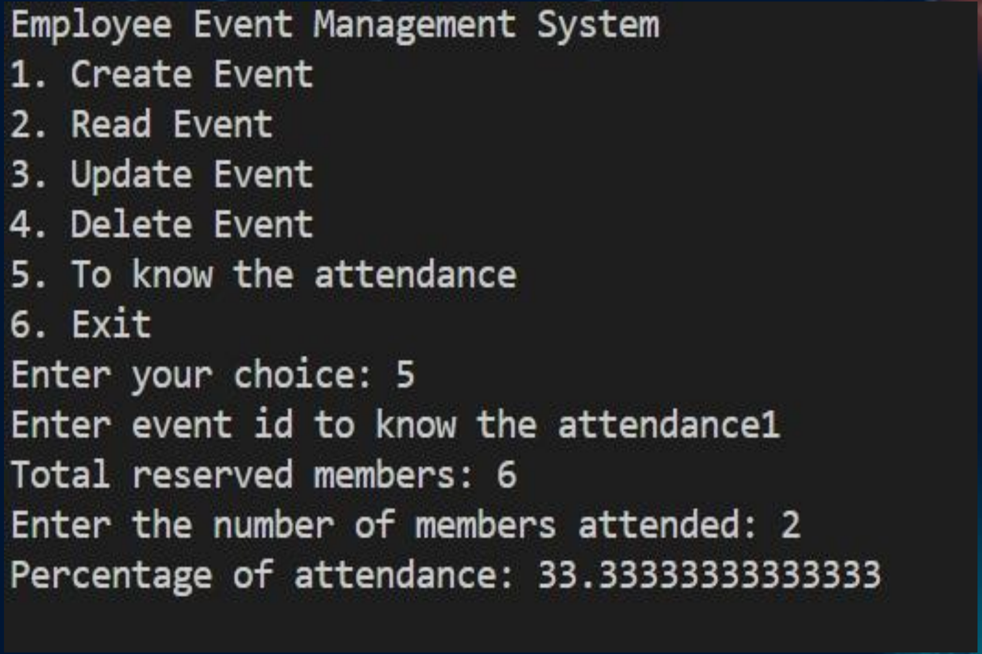
CHALLENGES

FUTURE SCOPE

OUTPUT



```
Employee Event Management System
1. Create Event
2. Read Event
3. Update Event
4. Delete Event
5. To know the attendance
6. Exit
Enter your choice: 1
Enter start date: 11-11-2023
Enter end date: 12-11-2023
Event ID: 1
Enter event title: INTERVIEW
Enter reserved members: 5
Event created successfully!
```



```
Employee Event Management System
1. Create Event
2. Read Event
3. Update Event
4. Delete Event
5. To know the attendance
6. Exit
Enter your choice: 5
Enter event id to know the attendance1
Total reserved members: 6
Enter the number of members attended: 2
Percentage of attendance: 33.3333333333333
```

CHALLENGES

- Input Validation
- Event Overlapping
- Attendance Calculation
- Complexity and Scalability
- Error Handling and User Experience

CONCLUSION

- The system supports basic operations like creating, reading, updating, and deleting events, allowing for efficient management of internal organizational events.
- Consider scalability aspects for larger event volumes and potentially add features such as reporting, or attendee management to enhance the system's capabilities.
- Strengthen the algorithm for detecting date conflicts, considering scenarios where events partially overlap or span across multiple days.

FUTURE SCOPE

- Allow events to be categorized or tagged, making it easier to filter and search for specific types of events. This can improve organization and retrieval of information.
- Make event fields customizable so that organizers can adapt the system to different types of events with varying requirements.

THANK YOU

ISMAIL D
AJEET KUMAR DANI
FARHAN
MANJUNATH SWAMY