A

SYNOPSIS REPORT

On

< Real-Time Event Scheduling System>

Submitted in partial fulfilment of the requirements of the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

Submitted to : -
Mr. Vipin Jadon
Assistant Professor
Department of
CSE,SOET

Submitted by:-
Manan Sahni (Roll No - BCAN1CA24088)
Mohd. Ziya Khan (Roll No -BCAN1CA24093)
Shreya Kumari (Roll No-BCAN1CA24148)

Date of Submission :- 12th March 2025

Department of Computer Science and Applications
School of Engineering and Technology
ITM University Gwalior, Madhya Pradesh
July 2024

Table of Contents

Group Details & Role of Members :-

Group Name :- **Timely Titans ⚡**

Topic :- Real-Time Event Scheduling System

1. Shreya Kumari (Roll no – BCAN1CA24148)

Project Manager & Researcher 📋
Mobile no – 9106538234

Email id – shreya12@gmail.com

2. Mohd. Ziya Khan (Roll no- BCAN1CA24093)

Developer & Data Structure Expert 💻

Mobile no – 8305687695

Email id – mohdziya144@gmail.com

3. Manan Sahni (Roll no – BCAN1CA24088)

UI/UX Designer & Tester 🎨🔍 Mobile no -9985674537

Email id – manan124@gmail.com

Project Based Learning

TITLE -: Real-Time Event Scheduling System Timely Titans ⚡

Data Structures (BCA-201)

Title 📅 of the Project:-

"SmartScheduler: A Real-Time Event Scheduling System"

Quote: ⏳ "Right time, right task—Smart scheduling for a smarter future!"

In today's fast-paced world, managing events efficiently is crucial for productivity and organization. **"SmartScheduler"** is a cutting-edge **Real-Time Event Scheduling System** designed to streamline the scheduling process by ensuring optimal time management. This system intelligently handles event creation, updates, and conflict resolution using advanced data structures such as **priority queues and heaps**.

With real-time updates, users can seamlessly schedule appointments, meetings, or tasks without overlapping issues. The system ensures that events are well-organized, accessible, and dynamically updated based on priority. By integrating smart scheduling algorithms, **"SmartScheduler"** enhances efficiency, making it a valuable tool for businesses, institutions, and individuals.

This project showcases the practical implementation of **data structures and algorithms**, contributing to real-world solutions in event management and scheduling systems. 🚀📅

Objective :-

✅ Efficient Scheduling: Ensure optimal event management by organizing tasks without conflicts.

✅ Real-Time Updates: Automatically adjust schedules and notify users of any changes dynamically.

✅ Priority-Based Allocation: Use priority queues or heaps to schedule events based on urgency.

✅ User-Friendly Interface: Provide a simple and intuitive UI for easy event creation and management.

✅ Conflict Resolution: Detect overlapping events and suggest the best available time slots.

✅ Scalability & Performance: Handle multiple users and large datasets efficiently.

✅ Automation & Notifications: Send automatic reminders and alerts for upcoming tasks or events.

✅ Data Storage & Retrieval: Use hash maps or databases to store and retrieve scheduling data quickly.

Scope :-

The **SmartScheduler** project has broad applications in various fields, ensuring efficient time management and real-time event handling. This system will improve **task scheduling, conflict resolution, and automated reminders** to enhance productivity.

1. Academic and Institutional Use 🎓

- Schools, colleges, and universities can use it for **exam timetables, class schedules, and faculty meetings**.
- Ensures no clashes in **room bookings** or **student-teacher schedules**.

2. Corporate and Business Applications 💼

- Organizations can manage **meetings, project deadlines, and employee task assignments**.
- Provides **automated alerts** for urgent events and prioritizes tasks accordingly.

3. Healthcare and Appointment Management 🏥

- Hospitals and clinics can schedule **doctor appointments, surgeries, and patient check-ups**.
- Reduces waiting time by efficiently allocating available time slots.

Process Description :-

The **SmartScheduler** system follows a structured and efficient process to manage real-time event scheduling. It ensures optimal time allocation, prevents scheduling conflicts, and provides automated notifications. The following steps outline the system's workflow:

1. User Registration & Authentication 🔐

- Users create an account and log in to access the scheduling system.
- Authentication ensures **secure access** and personalized scheduling.

2. Event Creation & Input Processing 📝

- Users enter event details such as **title, date, time, priority, and duration**.
- The system validates the input and checks for **time conflicts**.

3. Conflict Detection & Resolution ⚠

- The system compares the new event with existing schedules using **priority queues or hash maps**.
- If a conflict is detected:

  ✅ Suggests the nearest available time slot.

  ✅ Allows the user to **reschedule or override** based on priority.

4. Real-Time Scheduling & Optimization ⌛

- Uses **priority-based scheduling algorithms** (like heaps or queues) to organize events efficiently.
- High-priority events are scheduled first, ensuring **time-sensitive tasks** get precedence.

5. Notification & Reminder System 🔔

- Sends **real-time alerts** via email, SMS, or push notifications before event start time.
- Provides **reminders** for upcoming tasks to improve time management.

5. Event Modification & Deletion 🛠

- Users can **update, postpone, or delete** events.
- The system dynamically adjusts the schedule to maintain efficiency.

Resources & Limitations :-

Resources Required

To develop and implement the **SmartScheduler** system efficiently, the following resources are essential:

1. Hardware Resources 💻

- **Computer or Server** – For running the application and database.
- **Storage Devices** – For saving scheduled events and logs.
- **Internet Connectivity** – For cloud-based access and notifications.

2. Software Resources 🖥

- **Programming Languages** – Python, Java, or JavaScript for development.
- **Database Management System (DBMS)** – MySQL, PostgreSQL, or Firebase for event storage.
- **Development Frameworks** – Flask, Django (for web), or React, Angular (for UI).
- **Cloud Services** – AWS, Google Cloud, or Firebase for remote data storage and notifications.

3. Human Resources 👨‍💻

- **Developers** – Backend, frontend, and database experts.
- **UI/UX Designers** – To create an intuitive and user-friendly interface.
- **Testers** – For debugging and performance optimization.
- **Project Manager** – To oversee progress and ensure deadlines are met.

4. Additional Resources 🗄️

- **APIs & Libraries** – Integration of Google Calendar API, Notification APIs.
- **AI & Machine Learning (Optional)** – For automated scheduling recommendations.
- **Security Measures** – SSL encryption, authentication mechanisms, and backup systems.

💠 Limitations of the System

While **SmartScheduler** aims to provide seamless event scheduling, there are some challenges and constraints:

1. Complexity in Conflict Resolution ⚠️

- Managing multiple overlapping events efficiently requires advanced algorithms.
- Implementing AI-based smart scheduling might be **resource-intensive**.

2. Scalability Issues 📈

- Handling **large numbers of users and real-time events** may require **high server performance**.
- Cloud-based solutions may increase **operational costs**.

3. Dependency on Internet & System Resources 🌐

- A **stable internet connection** is required for real-time updates and notifications.
- **Slow or outdated devices** may impact user experience.

4. Security & Data Privacy 🔒

- **Unauthorized access** can lead to event data breaches.
- Ensuring **strong encryption and authentication** is necessary for security.

5. User Adoption & Learning Curve 📖

- Some users may find the interface **complicated** at first.
- Training or onboarding sessions may be required for **effective utilization**.

📌 Conclusion :-

The **SmartScheduler: A Real-Time Event Scheduling System** is a powerful tool designed to streamline event planning and prevent scheduling conflicts. By utilizing **priority queues, hash maps, and graph algorithms**, the system efficiently organizes tasks, optimizes scheduling, and sends real-time notifications. The project aims to enhance **productivity, time management, and automation** in various domains such as corporate meetings, academic schedules, and personal time management.

Ultimately, this project serves as a **practical and innovative** solution for individuals and organizations, enhancing efficiency and reducing the complexity of managing multiple tasks in real-time. ✅ 📆

📌 References :-

1. **Books & Research Papers** ○ Knuth, D. E. (1997). *The Art of Computer Programming: Sorting and Searching.* Addison-Wesley.
    - ○ Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms.* MIT Press.
2. **Online Articles & Websites** ○ GeeksforGeeks. “Priority Queue and Scheduling Algorithms.” Retrieved from www.geeksforgeeks.org
    - ○ IBM Cloud Blog. “Event-Driven Architecture for Real-Time Scheduling.” Retrieved from www.ibm.com ○ Stack Overflow Discussions on “Efficient Data Structures for Scheduling Systems.”
3. **APIs & Technology Documentation** ○ Google Calendar API Documentation: https://developers.google.com/calendar
    - ○ Firebase Cloud Messaging (for notifications): https://firebase.google.com/docs/cloud-messaging

📌 Bibliography :-

Author Contributions:

- **Project Lead** – Shreya Kumari
- **Development & Implementation** – Mohd.Ziya Khan
- **Testing & Debugging** – Manan Sahni **Sources Cited**:
- ChatGPT AI for structuring and writing content.
- Various research papers, websites, and books on **data structures, scheduling algorithms, and software architecture**.

This bibliography acknowledges all resources and contributors who helped in the research and development of **SmartScheduler: A Real-Time Event Scheduling System**. 📚✅