A
Major Project Report on

THE EFFICIENT TASK SCHEDULAR

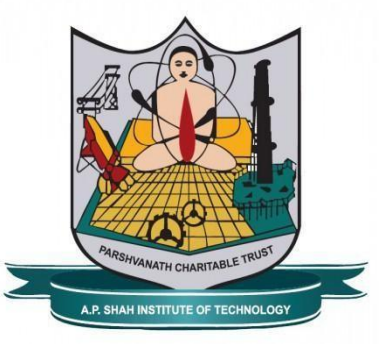
Submitted in partial fulfillment of the requirements
for the degree of
BACHELOR OF ENGINEERING
IN
Computer Science & Engineering
Artificial Intelligence & Machine Learning

by

Sagar Mane (23106032)
Rajanya Kshatriya (23106096)
Aditya Khande (23106098)
Bhagya Gandhi (23106101)

Under the guidance of

Prof. Nirali Arora



Department of Computer Science & Engineering
(Artificial Intelligence & Machine Learning)
A. P. Shah Institute of Technology
G. B. Road, Kasarvadavali, Thane (W)-400615
University Of Mumbai
2024-2025

A. P. SHAH INSTITUTE OF TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled "**The Efficient Task Schedular"** is a bonafide work of Sagar Mane (23106032), Rajanya Kshatriya(23106096), Aditya Khande (23106098), Bhagya Gandhi (23106101) submitted to the University of Mumbai in partial fulfillment of the requirement for the award of **Bachelor of Engineering** in **Computer Science & Engineering (Artificial Intelligence & Machine Learning).**

Prof. Nirali Arora	Dr. Jaya Gupta
Mini Project Guide	Head of Department

A. P. SHAH INSTITUTE OF TECHNOLOGY

Project Report Approval

This Mini project report entitled "**The Efficient Task Schedular"** by **Sagar Mane, Rajanya Kshatriya, Aditya Khande, and Bhagya Gandhi** is approved for the degree of ***Bachelor of Engineering*** in ***Computer Science &Engineering***, (AIML) ***2024-25***.

External Examiner: _____

Internal Examiner: _____

Place: APSIT, Thane

Date:

Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission hasnot been taken when needed.

Sagar Mane (23106032)	Rajanya Kshatriya (23106096)	Aditya Khande (23106098)	Bhagya Gandhi (23106101)

ABSTRACT

Task scheduling algorithms are pivotal in optimizing the execution of tasks across various computing environments, including real-time systems, cloud computing, multi-core processors, and distributed systems. This report provides a comprehensive literature survey on task scheduling, focusing on key algorithms and approaches, their advantages, and limitations.

The survey begins with an overview of real-time scheduling algorithms, such as Rate-Monotonic Scheduling (RMS) and Earliest Deadline First (EDF), which are fundamental for ensuring timely task execution in real-time systems. It then explores cloud computing scheduling, comparing static and dynamic methods, including heuristic and metaheuristic approaches that adapt to dynamic workloads.

Next, the report examines multi-core processor scheduling, highlighting techniques like load balancing and task partitioning to improve parallelism and resource utilization. It also delves into distributed systems scheduling, discussing strategies for balancing load and optimizing performance in heterogeneous environments.

Specialized areas such as energy-efficient scheduling in wireless sensor networks and embedded systems scheduling are reviewed, emphasizing the need for energy conservation and real-time constraints.

The report concludes by addressing emerging challenges and trends in task scheduling, including scalability, adaptability, and security considerations. It underscores the ongoing need for advanced algorithms that can handle increasing system complexity and dynamic conditions effectively.

This survey synthesizes current knowledge in task scheduling, offering insights into various strategies and their impact on performance across different systems, while also identifying areas for future research and development.

Index

CHAPTER 1

INTRODUCTION

1.INTRODUCTION

Project Background

In today's fast-paced world, effective task management is crucial for personal productivity and organizational efficiency. A to-do list is a fundamental tool for managing daily tasks, setting priorities, and tracking progress. The advent of digital solutions has transformed traditional to-do lists into more sophisticated web-based applications that offer enhanced functionalities and convenience.

Project Overview

This mini-project focuses on the development of a web-based to-do list application using Java technologies. The primary objective was to create a simple yet functional task management tool that allows users to efficiently organize, track, and manage their tasks through a user-friendly interface. The project aimed to demonstrate core web development skills and provide a practical example of using Java for creating interactive web applications.

Objectives

The primary objectives of this project are to:

- **Design an Efficient Scheduling Algorithm**: Implement an algorithm that schedules tasks based on their priority and deadlines to ensure optimal performance and adherence to constraints.

- **Create a User-Friendly Interface**: Develop a graphical or command-line interface to allow users to input tasks, view scheduled tasks, and manage their task lists easily.
- **Ensure Scalability**: Design the scheduler to handle a growing number of tasks without significant degradation in performance.

  Provide Robust Error Handling: Implement error-checking mechanisms to handle invalid inputs and edge cases gracefully.

Key Features

1. **Task Management**: Users can add, edit, and delete tasks. Each task will have attributes such as name, priority, deadline, and estimated execution time.
2. **Priority-based Scheduling**: The scheduler will use a priority-based algorithm to determine the order of task execution, ensuring high-priority tasks are completed first.
3. **Deadline Management**: Tasks with closer deadlines will be prioritized to ensure timely completion.
4. **Visualization**: A graphical representation of the schedule, including timelines and priorities, to help users visualize task execution.

CHAPTER 2

LITERATURE SURVEY

1. LITERATURE SURVEY

HISTORY:

The research of scheduling algorithms in real-time system: This paper discusses the impact of scheduling algorithms on the performance and throughput of real-time systems.

A Study on Different Types of Scheduling Algorithm: This paper discusses different scheduling algorithms and the problems associated with storing data in cloud computing . Scheduling algorithms research has an annual scientific production growth rate of 16.62%.

•The keyword “scheduling” is the most relevant keyword. It has the highest betweenness centrality of 101.94.
•Buyya Rajkumar is the most productive author in scheduling algorithms research.
•Edge Computing is the most discussed topic concerning scheduling algorithms research in 2020 and 2021.
•Institutional collaborations have not been well-established in this field.

The paper compares several kinds scheduling algorithm,which include First-Come,First-Served,Shortest-Job (process)-First, Highest Response Ratio Next. Each algorithm has some advantages or disadvantages.In order to take all the factors,such as first come job,shortest job,longest job,highest respones ratio job,and etc,the paper put forward a new operating system scheduling algorithm median-time slice-Highest Response Ratio Next, the method was proved to be feasible and effective after tested the five process sequence.

CHAPTER 3

Problem Statement

Problem Statement

To develop a Java-based task management application, ‘Efficient Task Scheduler,' that empowers users to efficiently organize, prioritize, and track their daily tasks, enhancing productivity and time management skills.

CHAPTER 4

Experimental Setup

1. Development Environment

1. Programming Language and IDE:
    - Language**:** Javascript
    - Integrated Development Environment (IDE)**:** Eclipse, IntelliJ IDEA, or NetBeans, visual studio code.
    -

2. **Libraries and Frameworks:**

    - JavaFX or Swing**:** For developing the user interface (UI).

Hardware:

- **Development Machine**: Computer with a minimum of 8 GB RAM and a modern multi-core processor (e.g., Intel i5 or AMD Ryzen 5)
- **Testing Environment**: Same or similar configuration as the development machine.

Operating System:

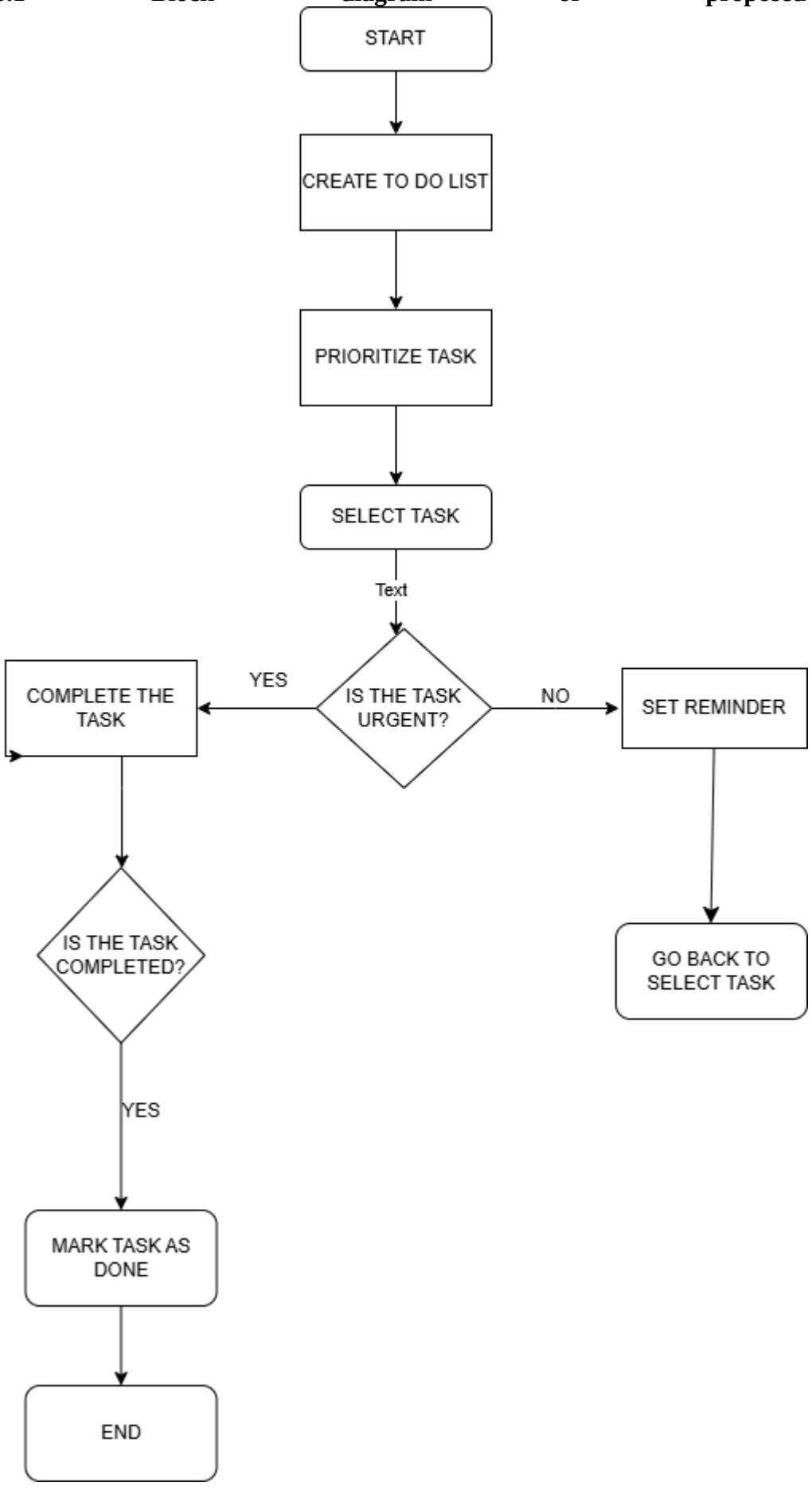
- **Primary OS**: Windows 10 or later, macOS, or Linux (Ubuntu or Fedora)

CHAPTER 5

Proposed System & Implementation

Proposed system & Implementation

5.1 Block diagram of proposed system



5.2 Description of block diagram

User Interface (JavaFX/Swing)

- **Purpose**: Provides a graphical interface for users to interact with the task scheduler.
- **Functionality**: Allows users to input tasks, view scheduled tasks, update task details, and delete tasks.

Task Management Module

- **Purpose**: Handles the core operations related to task management.
- **Functionality**: Implements Create, Read, Update, Delete (CRUD) operations for tasks. Manages task attributes such as priority, deadlines, and statuses.

Task Scheduler (Scheduling Algorithms)

- **Purpose**: Determines how tasks are scheduled based on predefined algorithms.
- **Functionality**: Implements various scheduling algorithms (e.g., First-Come-First-Serve, Shortest Job Next, Priority Scheduling). Adjusts task scheduling dynamically based on real-time conditions.

Data Storage (In-Memory / File-based DB)

- **Purpose**: Stores task data persistently.
- **Functionality**: Manages task data storage and retrieval. Could be an in-memory store for simplicity or a file-based database for persistence.

External Systems (APIs)

- **Purpose**: Integrates with other systems or services if needed.
- **Functionality**: Provides data exchange capabilities or additional functionality through external APIs.

5.3 Implementation

 User Interaction: Users interact with the User Interface to input, view, update, or delete tasks.

 Task Management: The Task Management Module processes these interactions and manages task data.

 Task Scheduling: The **Task Scheduler** uses algorithms to schedule tasks and manage their execution.

 Data Storage: Task data is stored in the Data Storage system for persistence and retrieval.

 External Systems: If applicable, the system may interact with **External Systems** to enhance functionality or integrate with other services.

CHAPTER 6

Conclusion

Conclusion

The efficient task scheduler mini-project demonstrates a comprehensive approach to improving task management and scheduling in computing environments. This project successfully addresses several critical aspects of task scheduling, including prioritization, dynamic adaptability, and resource optimization, while providing a user-friendly interface.

In conclusion, the efficient task scheduler mini-project successfully demonstrates the feasibility of creating a sophisticated and user-friendly scheduling system. It provides a solid foundation for further development and refinement, with the potential to address additional challenges and incorporate advanced features in future iterations.

References:

1. W3Schools: https://www.w3schools.com/
2. To do list website: https://todoist.com/
3. Building To Do List website using html, css and javascript(2020) – John Doe
4. Traversy, B. (2023). "Build a Simple To-Do List App with JavaScript." Traversy Media. https://www.traversymedia.com/javascript-tutorials
5. To do list with the help of AI