**Thành Viên   
Nguyễn Trung Tường 20225950  
Đặng Thành Tựu 20225951  
Nguyễn Sơn Tùng 20205044  
Thân Đức Vân 20225952  
Phạm Văn Tuyên 20225775**  
Problem statement  
 -Trong hệ điều hành, lập lịch CPU đóng vai trò quan trọng để đảm bảo rằng các tiến trình được thực thi một cách hiệu quả và tối ưu hóa thời gian xử lý

-Dự án này tập trung vào việc phân tích và so sánh các thuật toán lập lịch CPU như FCFS, SJNvà RR. Mục tiêu là đánh giá hiệu suất của từng thuật toán dựa trên các chỉ số như thời gian chờ trung bình (Average Waiting Time), thời gian đáp ứng (Response Time), và khả năng sử dụng CPU hiệu quả. Thông qua đó, người dùng có thể tham khảo để chọn thuật toán phù hợp cho hệ thống đa nhiệm   
**Use Case Diagram gồm các thành phần chính:**

2.1. Actor

**- User:** Người sử dụng hệ thống để mô phỏng và tính toán thuật toán lập lịch.

2.2. Use Cases Chính

-**Main Menu**:Giao diện chính nơi người dùng truy cập các chức năng.  
-bao gồm các Use Case:

+ **View Help**: Xem hướng dẫn sử dụng hệ thống.

+ **Exit Application**: Thoát khỏi ứng dụng (yêu cầu xác nhận qua Confirm Exit).

- **Input Processes**:Cho phép người dùng nhập thông tin các tiến trình.

-**Select Scheduling Algorithm:**Người dùng chọn thuật toán lập lịch

* FCFS
* SJN
* RR (thêm chức năng Set Time để đặt thời gian vòng quay).

**-Run Simulation:**Chạy mô phỏng các thuật toán lập lịch đã chọn.

Bao gồm:  
 +View Gantt Chart: Hiển thị biểu đồ Gantt.

+Calculator: Tính toán các chỉ số:

+Waiting Time (Thời gian chờ)

+Turnaround Time (Thời gian hoàn thành)

+Utilization (Mức sử dụng CPU).

**-View Processes**:Hiển thị thông tin các tiến trình đã nhập.  
 **-Display Result:**Hiển thị kết quả tính toán và mô phỏng.

3. Mối quan hệ Giữa Các Use Cases

* Các Use Cases chính như Main Menu, Run Simulation, và Select Scheduling Algorithm liên kết tới các Use Cases phụ thông qua quan hệ <include> hoặc <extend>.

**Class Diagram**   
1. Tổng quan hệ thống

-Sơ đồ lớp mô tả các thành phần chính trong hệ thống lập lịch CPU.   
Hệ thống này bao gồm các lớp có trách nhiệm thực hiện:

-Mô phỏng các thuật toán lập lịch như FCFS, SJN.

-Hiển thị kết quả, biểu đồ Gantt và các chỉ số hiệu suất như Thời gian chờ, Thời gian hoàn thành, và Mức sử dụng CPU.

-Tổ chức và điều khiển hoạt động của hệ thống.

2. Các lớp chính và chức năng  
  
2.1. Lớp Process

**Thuộc tính:**

processId, arrivalTime, burstTime, priority, waitingTime, turnaroundTime.

**Phương thức:**

* Process(): Khởi tạo thông tin tiến trình.
* getArrivalTime(), getBurstTime(), getPriority()...: Lấy thông tin thuộc tính tiến trình.
* setWaitingTime(), setTurnaroundTime(): Thiết lập thời gian chờ và thời gian hoàn thành.

2.2. Lớp GanttEntry

**Thuộc tính:**

* processId, startTime, endTime.

**Phương thức:**

* GanttEntry(): Khởi tạo Gantt Entry với thông tin tiến trình và thời gian.
* getStartTime(), getEndTime(): Lấy thời gian bắt đầu và kết thúc.

2.3. Lớp Scheduler

**Thuộc tính:**

* processList: Danh sách các tiến trình.

**Phương thức:**

* addProcess(): Thêm tiến trình vào danh sách.
* calculateMetrics(): Tính toán các chỉ số thời gian.
* generateGanttChart(): biểu đồ Gantt từ danh sách tiến trình.

**Lớp con kế thừa:**

* FCFSScheduler: biểu đồ Gantt và tính toán chỉ số theo thuật toán FCFS
* SJNScheduler: biểu đồ Gantt và tính toán theo thuật toán SJN
* RRScheduler:Thuộc tính: quantum.(Thêm tính toán cho thuật toán (RR).)

2.4. Lớp SimulationView

**Phương thức:**

* displayProcesses(): Hiển thị thông tin các tiến trình.
* updateGanttChart(): Cập nhật và hiển thị biểu đồ Gantt.
* displayMetrics(): Hiển thị chỉ số trung bình như thời gian chờ và thời gian hoàn thành.

2.5.HelpView

* showHelpInfo():hướng dẫn

2.6. Lớp MainMenuView

**Phương thức:**

* displayMenu(): Hiển thị menu chính.
* getAlgorithmChoice(): Lấy lựa chọn thuật toán từ người dùng.
* confirmExit(): Xác nhận thoát khỏi ứng dụng.

2.7SchedulerController:

Liên kết với lớp Scheduler để điều khiển mô phỏng thuật toán.

Các phương thức:

* setScheduler(): Thiết lập thuật toán lập lịch.
* startSimulation(): Bắt đầu mô phỏng.

2.8 MainController:

* Liên kết các thành phần: Menu, View và SchedulerController.
* Điều khiển luồng hoạt động của chương trình thông qua runApplication().

**3. Mối quan hệ giữa các lớp**

**Đóng gói:**

**-Các thuộc tính ở lớp Process được khai báo private**

**->bảo vệ dữ liệu, (getter setter) cho phép kiểm soát giá trị đầu vào đầu ra**

**Trừu Tượng**

**-lớp Scheduler định nghĩa Abstract, tạo bộ khung**

**- lớp FCFSScheduler, SJNScherduler, RRScheduler kế thừa**

**Đa Hình**

**-các lớp con FCFSScheduler, SJNScherduler, RRScheduler ghi đè overrider hai phuong thức calculateMetrics và generateGanttChart**

**Kế thừa**

**- lớp FCFSScheduler, SJNScherduler, RRScheduler kế thừa lớp cha Scheduler**

4. Kết luận

Sơ đồ lớp trên đã mô tả chi tiết các thành phần chính của hệ thống lập lịch CPU. Hệ thống đảm bảo tính mô-đun hóa, với sự phân tách rõ ràng giữa:

* Dữ liệu (Process, GanttEntry).
* Xử lý (Scheduler và các lớp con).
* Hiển thị (SimulationView, MainMenuView).
* Điều khiển (Controller).