

# CPU Scheduling Simulator

박상호  
이영섭  
(발표자) 임정섭  
진시윤

# INDEX

- I. 프로그램 소개
- II. 프로그램 구조
- III. 프로그램 장점 및 특징
- IV. 개발 과정
- V. 느낀점

## I. 프로그램 소개

개발언어 - Python

| Python



| PyQt5



| Plotly



## I. 프로그램 소개

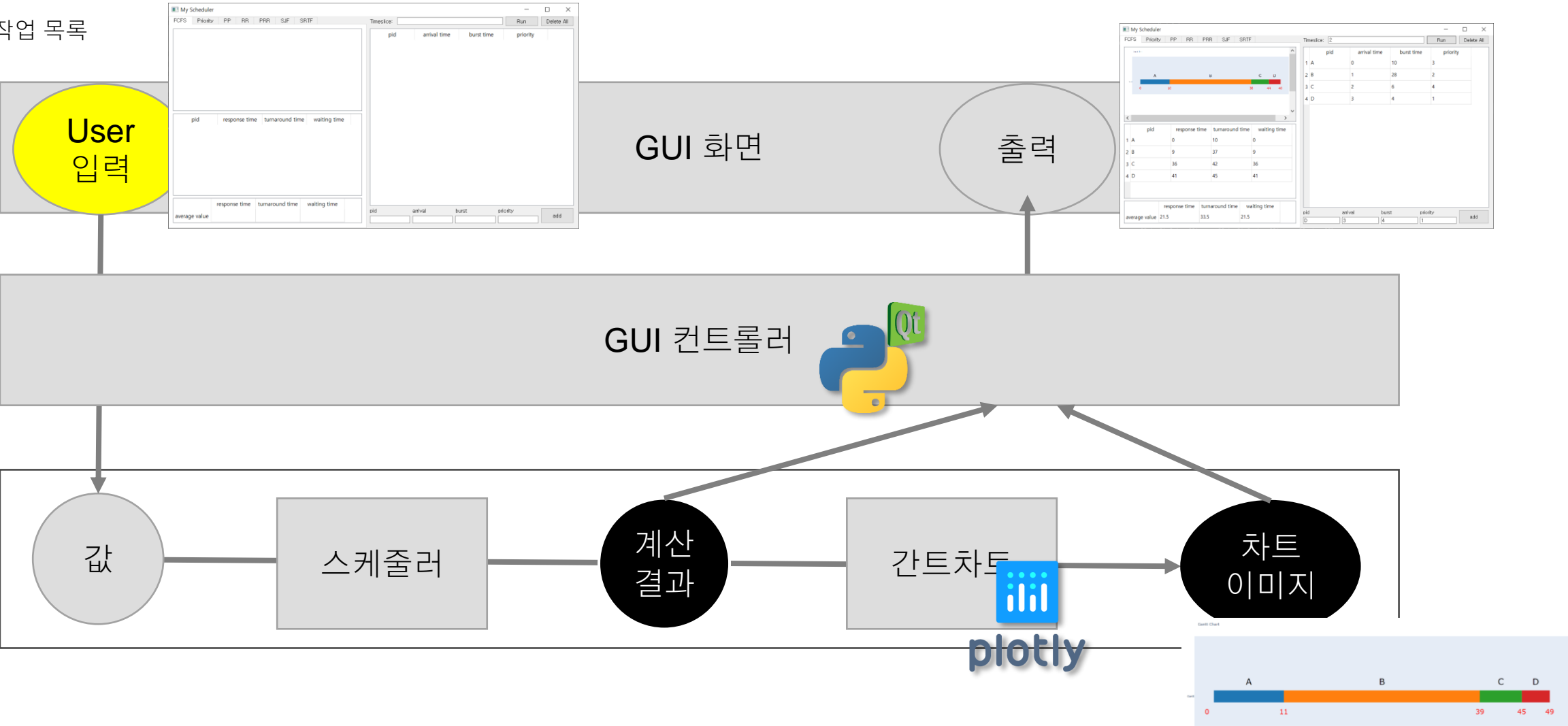
### 개발 작업 목록

1. 스케줄러(FCFS, RR, Priority, Preemptive Priority, Priority RR, SJF, SRTF)
2. 간트 차트
3. GUI
  - GUI 화면 디자인
  - GUI 컨트롤러



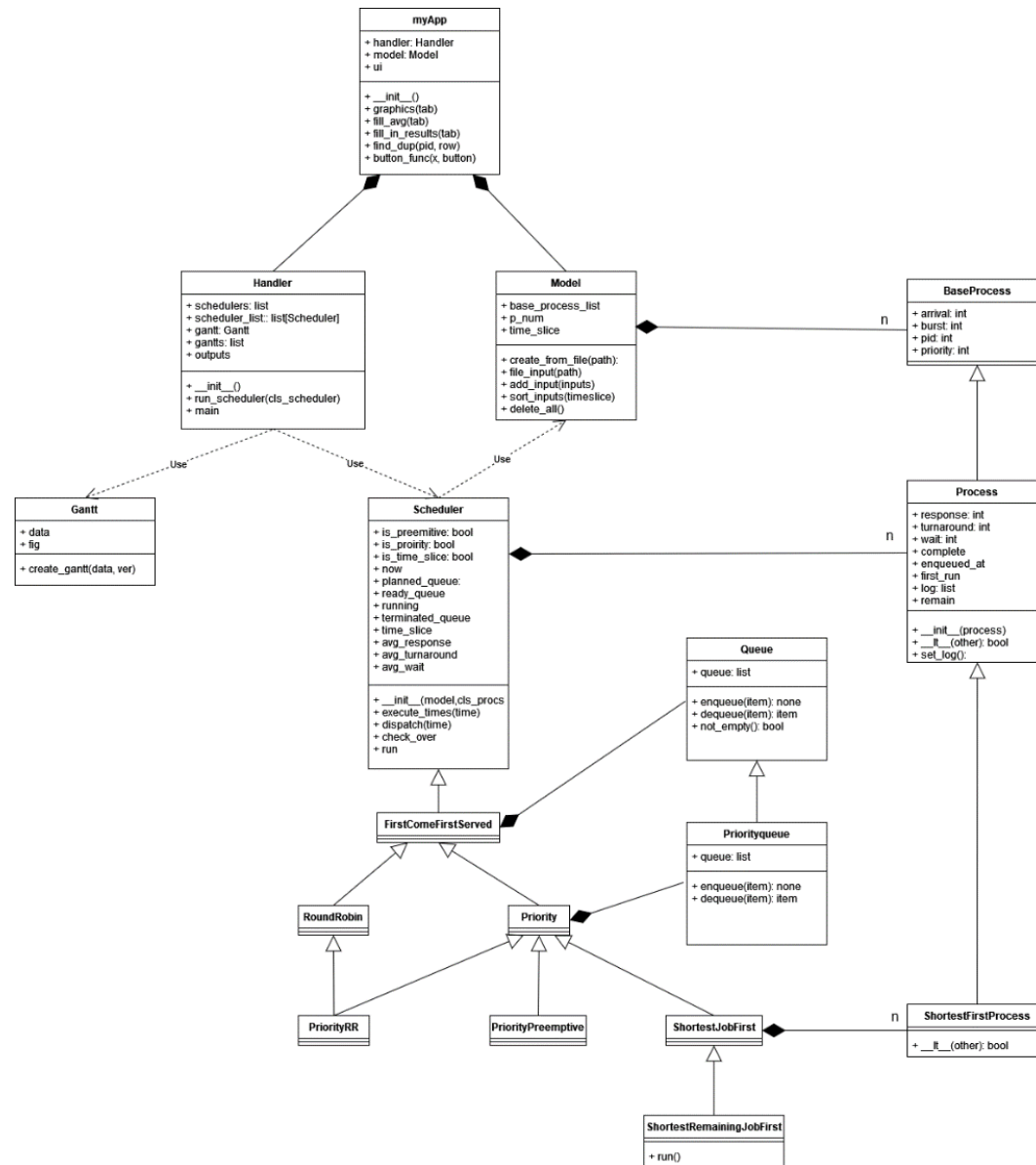
I. 프로그램 소개

개발 작업 목록



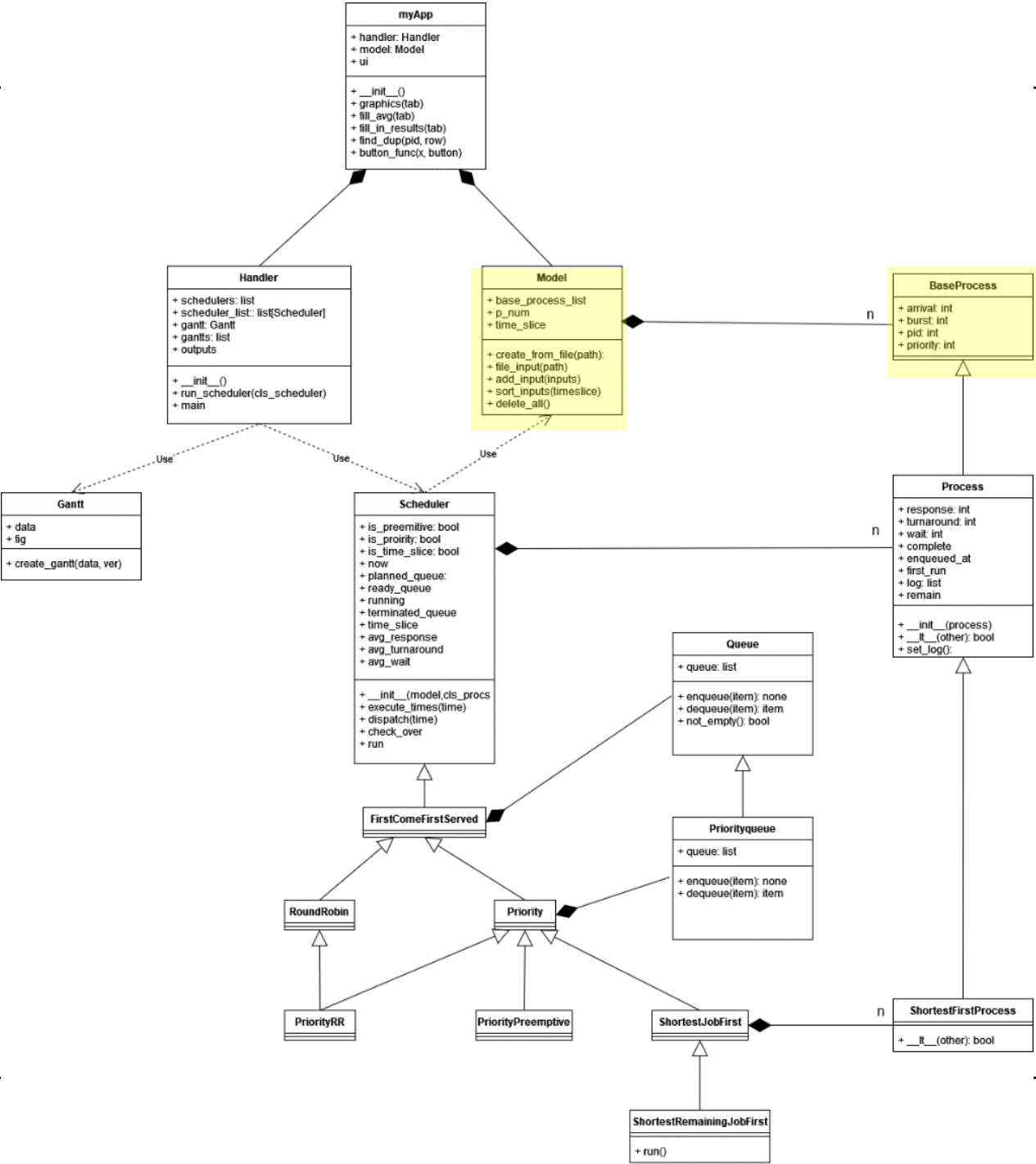
## II. 프로그램 구조

### 클래스 다이어그램



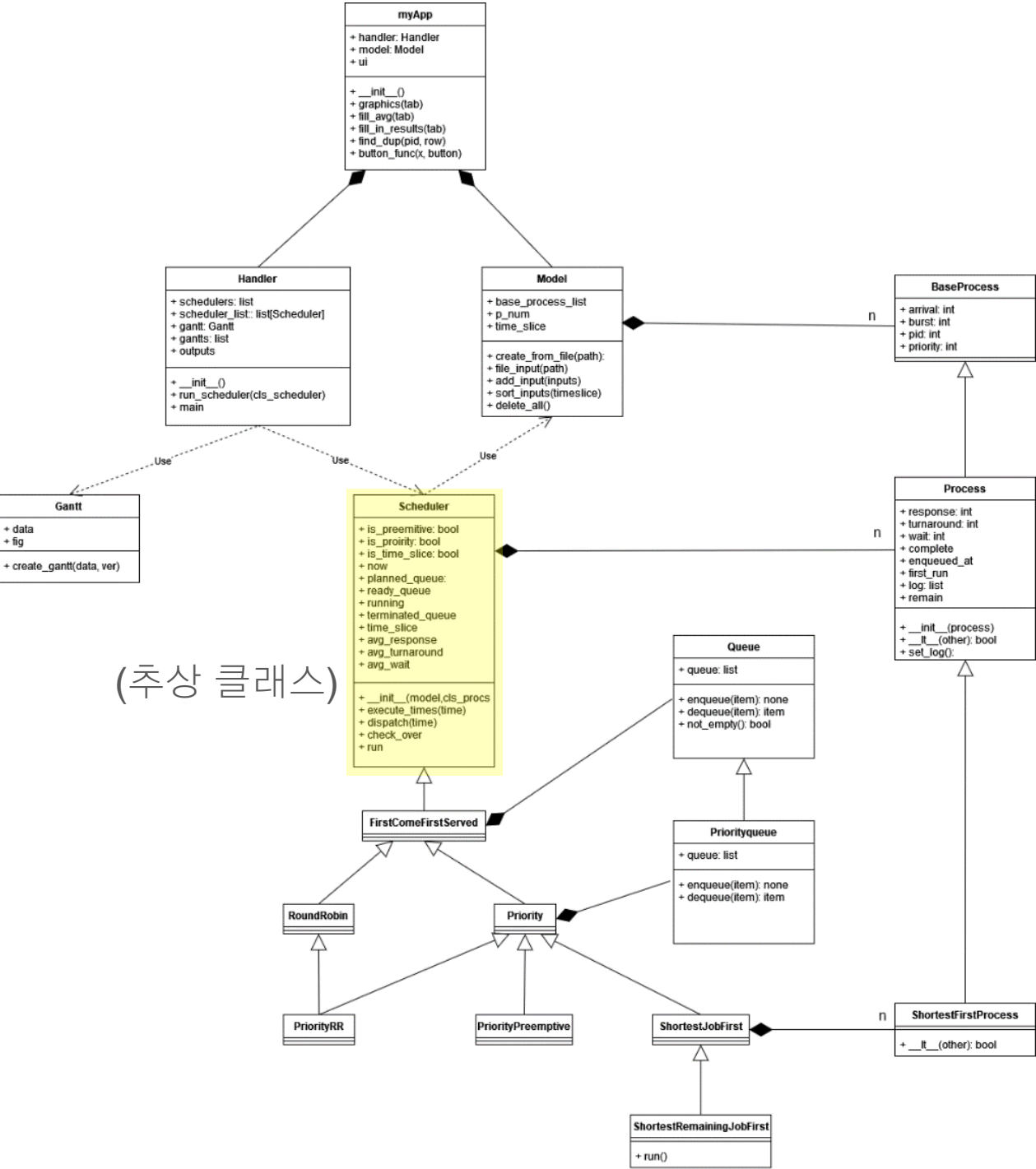
II. 프로그램 구조

클래스 다이어그램



II. 프로그램 구조

클래스 다이어그램





```
class FirstComeFirstServed(Scheduler):  
    is_preemptive = False # 실행 중간에 프로세스 교체 허용?  
    is_priority = False # ready queue가 priority queue or FIFO queue  
    is_time_slice = False # time slice 적용?
```

```
class Priority(FirstComeFirstServed):  
    is_priority = True
```

```
class PriorityPreemptive(Priority):  
    is_preemptive = True
```

```
class RoundRobin(FirstComeFirstServed):  
    is_time_slice = True
```

```
class PriorityRR(Priority, RoundRobin):  
    pass
```

```
class ShortestJobFirst(Priority):  
    process_class = ShortestFirstProcess # 프로세스 객체 안 바꾸면 heap에서 우선순위 비교 시 오류 발생
```

20

```
def run(self):
    while self.check_over():
        # save first_run time
        if self.running.first_run is None:
            self.running.first_run = self.now

        next_dispatch_time = self.running.remain
        if self.is_time_slice and self.time_slice < self.running.remain:
            next_dispatch_time = self.time_slice

        if self.planned_queue:
            next_arrival_time = self.planned_queue[0].arrival - self.now
            if next_arrival_time <= next_dispatch_time:
                # arrive
                new_process = self.planned_queue.pop(0)
                self.ready_queue.enqueue(new_process)
                # preempt by priority
                if self.is_priority and next_arrival_time == 0 and new_process < self.running:
                    self.dispatch()
                    if self.running.first_run == self.now:
                        self.running.first_run = None
                elif self.is_preemptive and new_process < self.running:
                    self.dispatch(next_arrival_time)
                continue

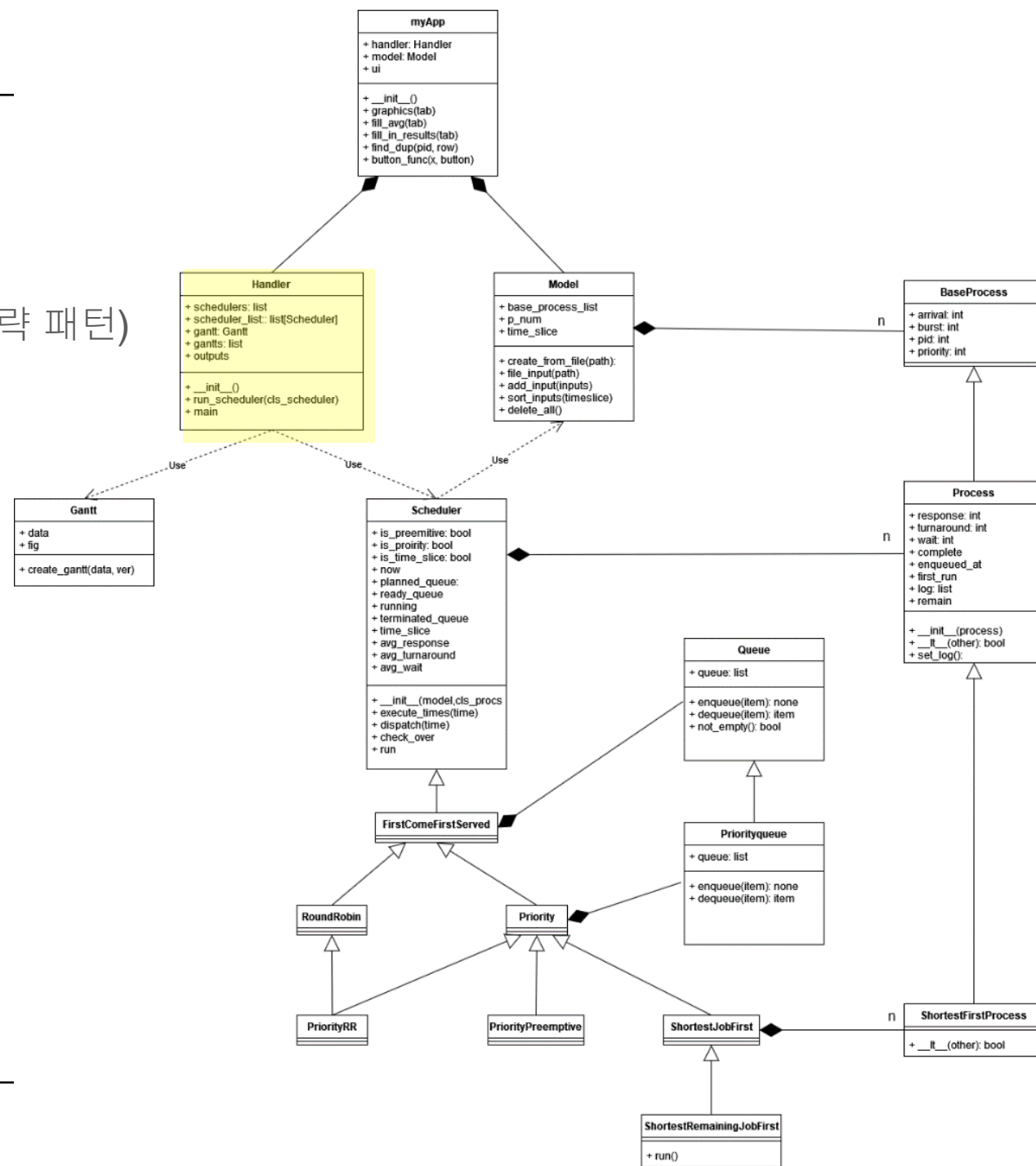
        self.dispatch(next_dispatch_time)
```

20

## II. 프로그램 구조

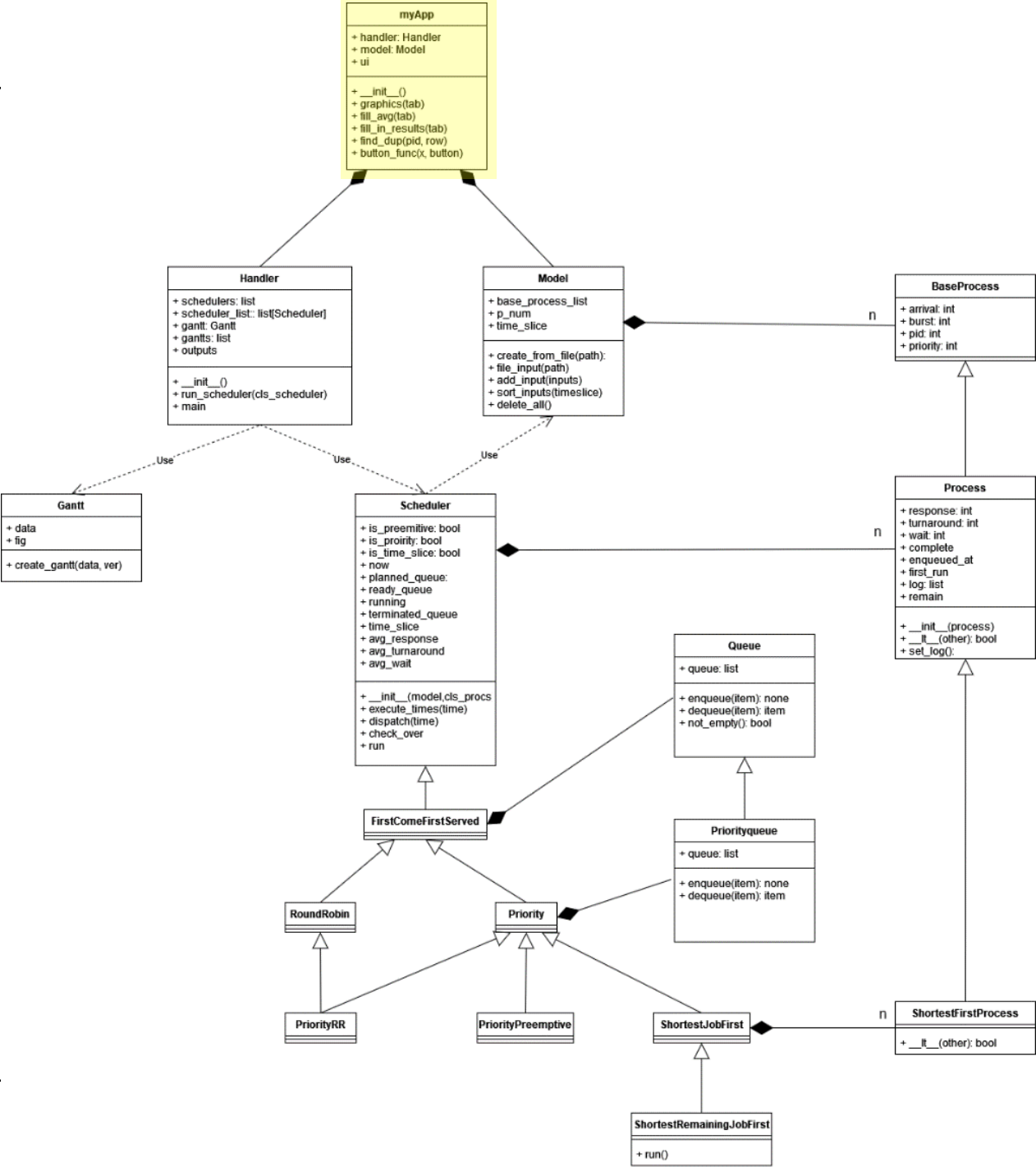
## 클래스 다이어그램

(전략 패턴)



II. 프로그램 구조

클래스 다이어그램



### III. 프로그램 장점 및 특징

#### 장점 및 특징

##### 확장성

- 객체지향: 추상 클래스 및 전략패턴을 적용하여 스케줄러를 구현했으므로 사용자가 스케줄링 알고리즘을 쉽게 추가할 수 있다.

##### 사용성

- GUI 제공: 사용자가 눈으로 입력 및 출력을 함
- UX: 스케줄링 계산은 모든 알고리즘에 대해 한꺼번에 진행됨  
출력화면에 탭을 적용하여 여러 알고리즘 간의 출력 전환이 간편함
- Validation: 잘못된 입력에 대해 거부 및 경고 메시지 출력

## IV. 개발과정

팀원역할 정하기 - 팀원역할

### | 박상호

보고서 작성,  
PPT 작성

### | 이영섭

보고서 작성, UI  
설계, UI 개발

### | 임정섭

프로세스, 클래스 및  
스케줄러 구성, 전반  
적인 개발 프로세스  
관리 및 조율

### | 진시윤

간트차트 구현,  
UI 설계, UI 개발,  
GUI 프로그래밍

IV. 개발과정

과정소개 - Notion

Tasks

보드 · 상태별

보드 · 담당자별

내 작업

모든 작업

캘린더

보기 추가

필터

정렬

...

새로 만들기

미린 업무 2

IDLE 프로세스 고려

임정섭

변수명 변경

임정섭

+ 새로 만들기

진행 예정 0

+ 새로 만들기

진행 중 3

[PyQt] 결과 시간 출력

진시윤 youngseob lee

3

[PyQt] 실행 파일 만들기

진시윤 youngseob lee

발표 준비

임정섭 youngseob lee 진시윤 호 상

2022년 6월 3일

+ 새로 만들기

완료 15

SRTF 제작

임정섭

GUI 기획

임정섭 진시윤 youngseob lee 호 상

2022년 5월 20일

계획서 작성

youngseob lee

2022년 5월 19일

PyQt 사용 결정

youngseob lee 진시윤 호 상

2022년 5월 18일

FCFS 알고리즘 제작

임정섭 진시윤

2022년 5월 17일

입력 모듈 제작

임정섭 진시윤

2022년 5월 17일

회의록

모든 회의

...

새로 만들기

필터

정렬

...

새로 만들기

[5/30] GUI 점검 및 발표 준비

주간 회의

정섭 호

2022년 5월 30일 오후 10:01

[5/19] GUI 사용 결정 및 계획서 작성

주간 회의

정섭 호

2022년 5월 19일 오후 11:59

[5/16] 계획서 작성

주간 회의

정섭 호

2022년 5월 16일 오후 7:27

[5/12] 코드 설명 및 역할 분담

임시 회의

정섭

2022년 5월 12일 오후 7:15

[5/10] 일정 계획 및 모듈 설계

주간 회의

정섭 호

2022년 5월 10일 오후 10:31

[5/6] 첫 회의

주간 회의

정섭 호

2022년 5월 10일 오후 10:29

+ 새로 만들기

## IV. 개발과정

### 과정소개 - Notion


[도움필요한 부분]

4, 5번 도움 주실 수 있는지..


4) 현재 간트 차트를 `plotly.figure_factory`로 만들어 놓았는데, 이를 저장할 방법을 찾아보아야 할 듯해요.

5) 번은 사실 그냥 계산하는 방법으로 할지 아니면 코드 내에서 자체적으로 계산한 값을 가져오는 방법으로 할지..


+ 정섭님 SRTF는 일단 배제하고 했습니다!

 임정섭 5월 25일


4번은 시간 있을 때 함께 찾아보도록 해요. 5번은 scheduler 클래스에 프로퍼티로 `avg_turnaround`, `avg_response`, `avg_wait` 메서드를 구현해놓을테니 가져다 쓰시면 될 거 같아요.

 임정섭 5월 25일


제가 지금 바빠서 코딩이랑 깃허브 푸시는 금요일 2시 전까지 해놓겠습니다.

 임정섭 5월 25일

금요일 오전쯤에 회의 시간 갖고 플젝 점검 해보죠

 임정섭 5월 25일

음 아니면 아예 `turnaround`, `response`, `wait times dictionary`를 프로퍼티로 넘겨줘도 괜찮을 거 같네요. `avg_` 메서드를 만드는 건 너무 GUI 프로그램에 종속적인 느낌이 들어서요,

 임정섭 5월 25일

구현은 어렵지 않으니, 나중에 회의해서 결정합시다


 진시윤 5월 26일

4) `Qgraphicsview`에 이미지 파일 업로드하는 방법은 확인했지만, 간트차트를 이미지로 저장하는 방법 고민중입니다.

 진시윤 5월 11일

++ 추가사항 : pid는 반드시 추가!!!해야 할 것 같고요, quantum은 어떻게 할까요


+++ 위 이슈에 대해 생각난 사항 : arrival time이나 priority 순으로 sort하는 알고리즘 하나 구현하는 건 어떨까요 ( + 이진 큐와 관련지으면 될 듯한데, 처음에만 sort하면 될 듯합니다)

 임정섭 5월 12일

- pid 추가하도록 하겠습니다.

- quantum(slice)는 코드 상에서 Model이라는 객체를 정의해서 해결했어요. Model은 process의 list, time slice, 프로세스 개수(p\_num)을 포함하는 자료구조입니다.

- 네 알고리즘 코드 고민해보니 arrival time 순으로 정렬하는 게 나을 거 같아요.

 임정섭 5월 12일

스케줄링 모듈에서는 프로세스를 세 자료구조에 보관하면 될 것 같습니다.


1. 아직 도착하지 않은 프로세스 - order by arrival\_time asc

2. ready queue - FCFS/RR은 큐, 나머지는 heap 사용

3. 실행 중인 프로세스 - 그냥 변수로 저장

 시윤 진 5월 12일

네네 저도 그렇게 생각해요

 임정섭 5월 12일

프로세스 객체는 그냥 글로벌하게 하나 만들어놓고 스케줄러 들어갈 때마다 remain, complete, first run time을 초기화해서 사용하려고 합니다.

 임정섭 5월 12일

아니면 대안이 두 가지가 있어요. 본문에 적겠습니다



## IV. 개발과정

### 과정소개 - Slack

#### ▼ 채널

# code-reference

# code-review

# feature

# general

# random

# report

# tools

+ 채널 추가

임정섭 오후 11:24

**31** 운체 팀플 회의  
수 6월 1 at 10:30 - 11:30 오후

5명의 게스트: 2 예, 3이(가) 응답하지 않음

사용 가능한 미팅 링크 없음

이동할까요? [예](#) [아니요](#) [미정](#)

이영섭 오후 11:25

\$ git checkout -t origin/feature/create-meeting

임정섭 오후 11:26

**31** 운체 팀플 발표 준비  
금 6월 3 at 11:00 - 12:30 오후

5명의 게스트: 1 예, 4이(가) 응답하지 않음

사용 가능한 미팅 링크 없음

이동할까요? [예](#) [아니요](#) [미정](#)

## V. 느낀 점

Q. 개발하면서 즐거웠던 점이 있었는가?

팀장님과 팀원님이 일정이 뒤쳐지지 않게끔 항상 잘 이끌어 주셔서 팀 프로젝트의 이점을 경험할 수 있었다.

의견을 교류하고 피드백을 하는 활동을 동반한 많은 회의를 통해 열심히 할 수 있는 동기부여가 되었다.

## V. 느낀 점

Q. 개발하면서 힘들거나 아쉬웠던 점이 있었는가?

- 개발과정 막바지에 **pyinstaller**를 활용하여 실행 파일을 제작하는데 어려움을 겪었다.
- 고려해야 할 점으로 **module** 설치 시 **conda install**이 아닌 **pip**를 활용하여야 한다는 점, **spec file**에 **datas**에 **ui files**를 추가해야 한다는 점, **Explicit import**가 아닌 **numpy**와 **pandas module**을 **spec file**의 **hidden\_import**에 추가해야 한다는 점 등이 있었다.
- **github conflict** 때문에 어려움을 겪었다.
- 개발 과정 시 **pull**하는 행동이 습관화되어 있지 않아서 **file conflict**가 빈번하게 발생하여 해결하는 데 어려움을 겪었다.

## V. 느낀 점

Q. 팀원에게서 배운 점은?

notion, slack, github review, branch 등 일정과 코드관리를 위해 다양한 tool을 사용하는 방법을 배웠다.

Pyqt5, pyinstaller 등 다양한 패키지를 사용하는 방법을 배웠다.

추상과 상속, 모듈화를 활용하며 객체지향 개념에 대해 더 깊게 이해할 수 있었다.

**Thank U and QnA**