

# UI 설계 (초기 화면)

스케줄러

음력

< 2018 ▾ 12 ▾ >

월	화	수	목	금	토	일
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Title:

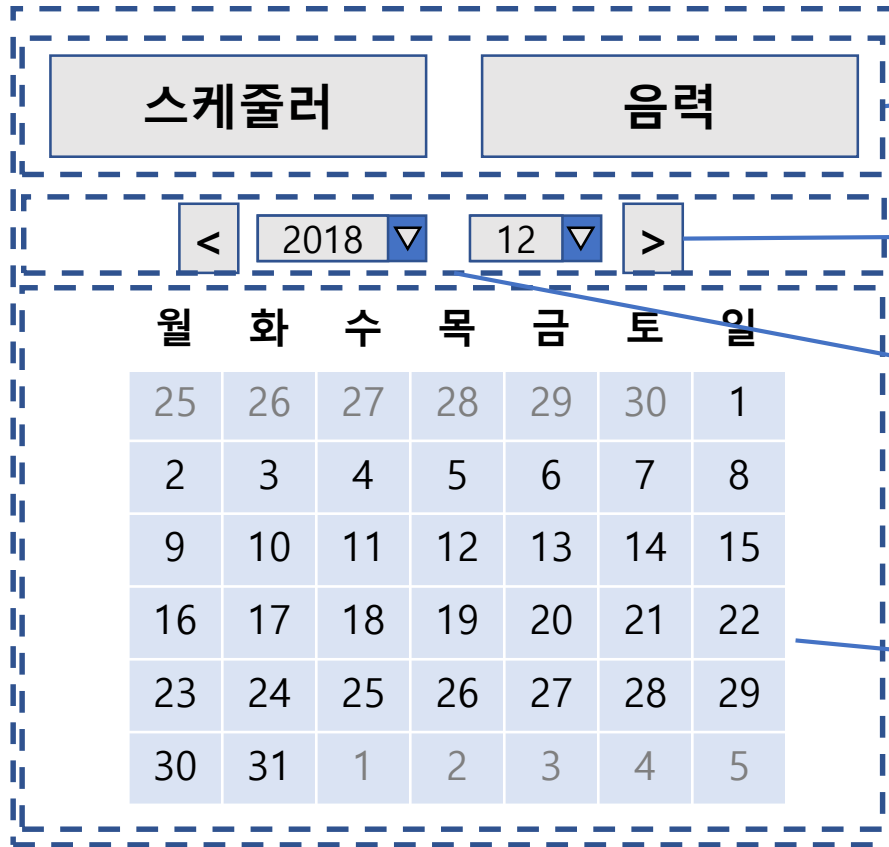
Place:

Time:  ~

Content:

Modifying

# UI 설계 (좌측 레이아웃)



**Widget:** Button

**기능:** 클릭 시 main Layout 우측의 탭을 바꾸어 준다.

**Widget:** Button

**기능:** 클릭 시 이전 혹은 다음 달로 넘어간다. 1월일 경우 이전 해 12월, 12월인 경우 다음 해 1월로 넘어간다.

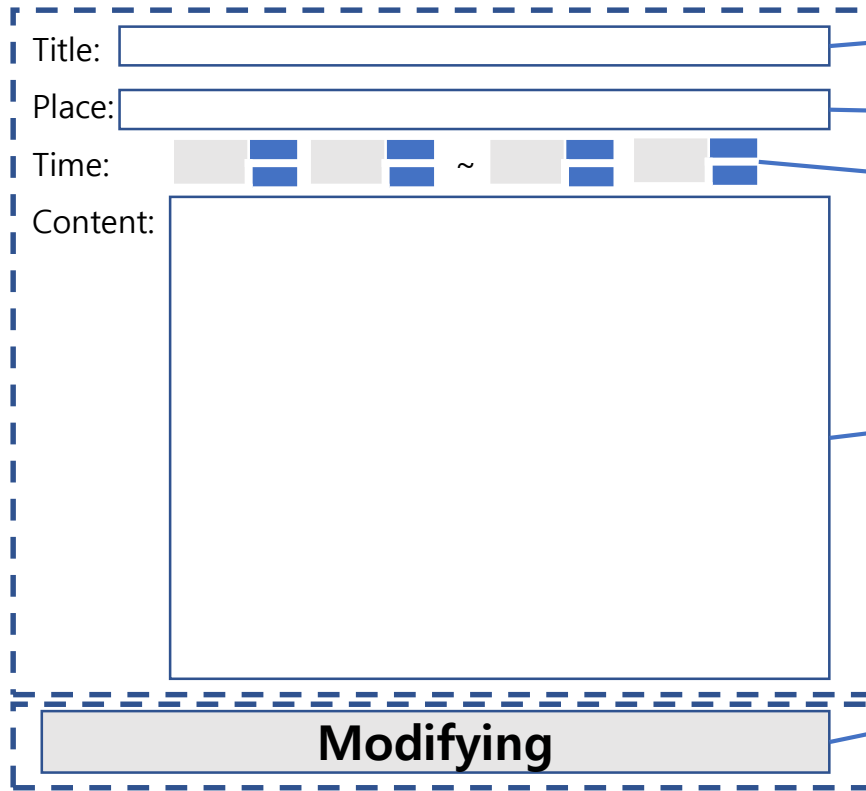
**Widget:** Combo Box

**기능:** 특정 연도, 월로 이동한다. 년의 경우 (1980 ~ 2040), 월의 경우 (1 ~ 12) 를 범위로 잡는다.

**layout:** Grid, **Widget:** Button

**기능:** 특정 연도, 월에 따른 달력을 출력한다. 휴일일 경우 일자가 붉게 표시된다. 일정이 있을 경우, 버튼이 푸른색으로 표시된다.

# UI 설계 (우측 레이아웃, 스케줄러)



The diagram illustrates a user interface for a scheduler, enclosed in a dashed blue border. It features four input fields: 'Title:', 'Place:', 'Time:', and 'Content:'. The 'Time:' field is represented by a sequence of gray squares, each containing a blue equals sign, separated by a tilde (~). Below these fields is a large white rectangular area for content. At the bottom, a gray button labeled 'Modifying' is positioned. Blue lines connect each input field and the button to their respective descriptions on the right.

**Widget:** label, LineEdit  
**기능:** 일정의 제목을 입력 받는다.

**Widget:** label, LineEdit  
**기능:** 일정의 장소를 입력 받는다.

**Widget:** label, Spinner  
**기능:** 일정의 시간을 입력 받는다.

**Widget:** label, TextEdit  
**기능:** 일정의 내용을 입력 받는다.

**Widget:** Button  
**기능:** 일정의 내용을 저장한다.

# UI 설계 (우측 레이아웃, 음력)

오늘의 날짜정보

양력 날짜

20XX-XX-XX

음력 날짜

20XX-XX-XX

양력 -> 음력

Convert

Reset

**Widget:** label

**기능:** 탭 이동시 등장, Convert 버튼 입력 시 사라지고, Reset 버튼을 입력하면 다시 나타난다.

**Widget:** label

**기능:** 변환된 양 / 음력 날짜가 출력된다.

**Widget:** Spinner

**기능:** 변환할 연도, 월, 일을 입력 받는다.

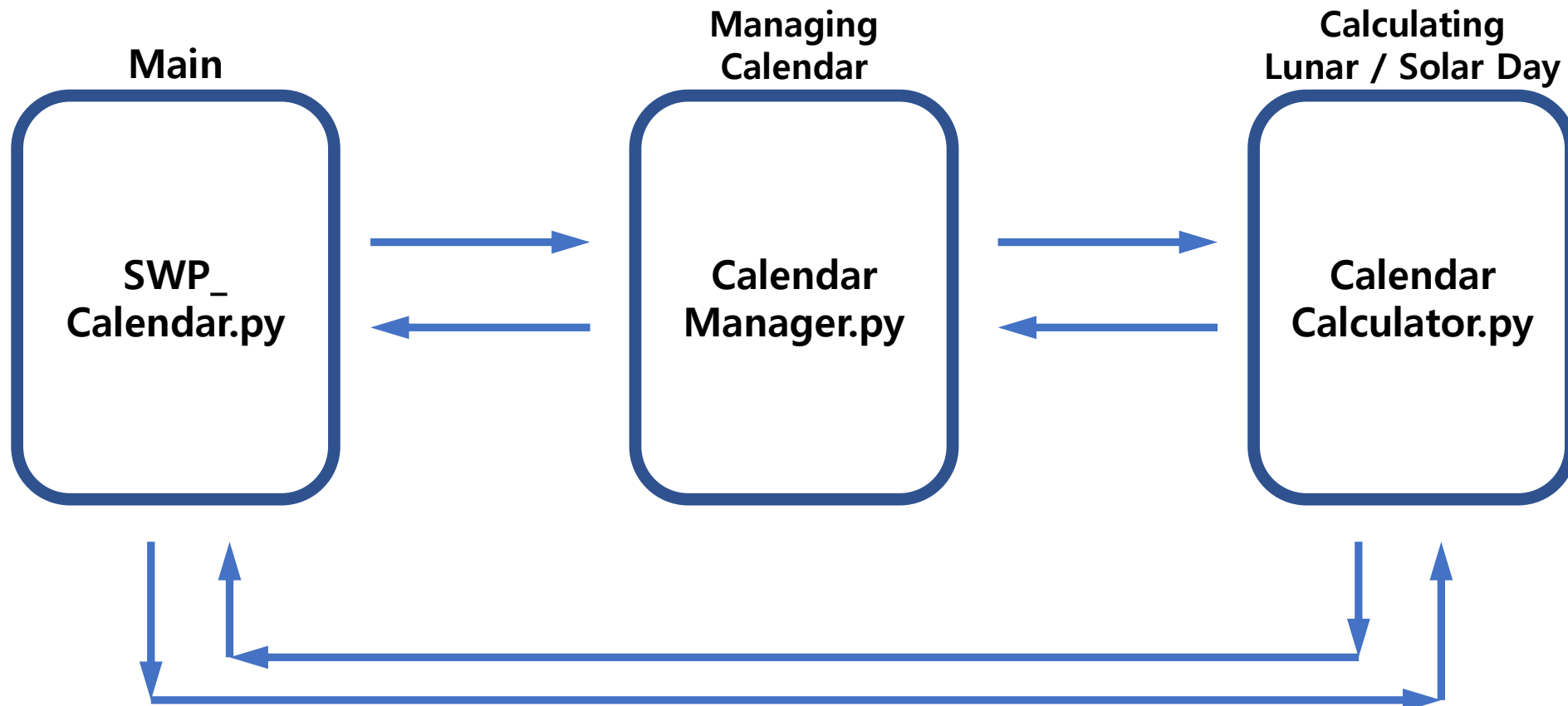
**Widget:** combo Box

**기능:** 양력에서 음력으로 혹은, 음력에서 양력으로 변환할 지를 선택한다.

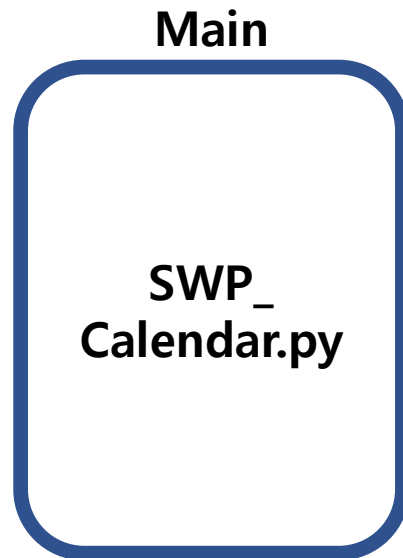
**Widget:** Button

**기능:** Convert 버튼을 사용할 경우, combo box의 내용에 따라 양 / 음력을 변환한다. Reset 버튼을 사용할 경우, 현재의 날짜 정보(양력 + 음력)를 출력한다.

# 소프트웨어 구조 설계 (전체 구조)



# 소프트웨어 구조 설계 (전체 구조)



GUI를 구성하고 메인 창을 띄운다. 일정을 pickle로 저장한다.  
달력 연산 외의 모든 처리를 진행한다.  
calendarManager로부터 달력 정보를 받아 달력을 출력한다.  
calendarCalculator에게 연도, 월, 일을 보내고 음/양력 날짜를 받는다.

- renderDate()
- btnEvent()
- Modifying()
- previousMonth()
- nextMonth()
- selectionChanged()
- clearLayout()
- setScheduleUI()
- setLunaDateUI()
- display().
- lunarBtnEvent()

# 소프트웨어 구조 설계 (전체 구조)

Managing  
Calendar

calendar  
Manager.py

달력 정보를 연산한다. 일정을 클래스에 저장한다.

입력 받은 년, 월, 일로 2차원 배열을 생산한다. 각 요소는 1부터 30 혹은 31로 구성되어 있으며, 각 요소의 인덱스 값은 달력의 좌표 값을 나타낸다.

휴일들을 튜플로 저장한다. Ex. -> (월, 일, 휴일 명)

일정을 내부 클래스에 저장한다. 이 클래스는 pickle로 저장된다.

- `getCalendar()`
- `getMaxDay()`
- `setYear()`
- `setMonth()`
- `setCalendar()`
- `makeCalendar()`
- `enrollHoliday()`
- `loadHoliday()`

# 소프트웨어 구조 설계 (전체 구조)

Calculating  
Lunar / Solar Day

calendar  
Calculator.py

음력과 양력을 연산한다

입력 받은 년, 월, 일을 음력 혹은 양력으로 변환한 뒤 반환한다.  
음력 휴일을 계산하여 튜플로 반환한다. Ex. -> (월, 일, 휴일 명)

- isLunarMonth()
- toSolarDate()
- toLunarDate()
- formatDate()
- getSolarHoliday()
- getToLunarDate()