

Assignment #2

Overview Documentation

2020095178 최윤선

✓ Metadata

Family Calendar 스키마 구현 및 Java UI 구현

✓ Summary

- 과제에서 가장 어려웠던 부분
 - Java로 UI를 구현하는 것이 가장 복잡했고, 상당히 많은 시간을 할애하였다.
 - 달력의 수많은 기능을 생각하며, relation을 따지고 스키마를 짜는 데에도 많은 고민을 하였다.
- 디자인에 관한 결정
 - Requirements를 최대한 반영하기 위한 UI를 디자인하기 위해 노력하였다. Google Calendar 웹사이트를 보며 공유 캘린더의 구동과 디자인을 참고하였다. 또한, 구글링을 통해 찾은 가장 기본적인 Calendar UI를 바탕으로 버튼과 기능을 하나씩 추가해가며 구현하였다.
- 프로그램을 설계, 구현 또는 테스트하는 동안 발생한 주요 문제
 - Calendar UI에 month, week, day를 바꿀 수 있는 이전/다음 버튼을 삽입하였는데, 처음 테스트 때 이 버튼이 잘 작동되지 않아, 많은 수정을 거쳤다.

✓ Specification

- Version: Ubuntu == 22.04.2 / PostgreSQL == 14
- UI 구현 및 스키마 작성을 위해 참고한 9 가지 Requirements
 - Book an event
 - Check availability of users
 - Invite users to the meeting/event
 - Request RSVP
 - Meeting Reminders
 - Modify the meeting /event
 - Cancel a meeting/event
 - Lookup for all meetings in the calendar
 - View Calendar

✓ Design and Implementation

· Java UI 및 기능

FamilyCalendar.java, WeekCalendar.java, DayCalendar.java 이렇게 세 개의 파일로 UI 를 구성하였으며, 세 개의 java 파일은 서로 상속 관계이다.

- FamilyCalendar.java

버튼을 클릭했을 때, 발생하는 action 을 아래의 함수 안에 모두 구현하였다.

```
123      @Override
124      public void actionPerformed(ActionEvent e){
125          Object obj = e.getSource();
126      }
```

Day/Week 버튼을 눌렀을 때, Daily/Weekly Calendar 로 넘어가도록 구현

```
168          if(obj == dayBtn){
169              if(loginBtn.getText().equals(anObject:"Logout")){
170                  new DayCalendar();
171                  setVisible(b:false);
172              }
```

```
178          if(obj == weekBtn){
179              if(loginBtn.getText().equals(anObject:"Logout")){
180                  new WeekCalendar();
181                  setVisible(b:false);
182              }
```

로그인 창 구현

```
198      public void loginWindow(){
```

Notification 창 구현

```
240      public void notiWindow(){
```

일정 추가 창 구현

```
258      public void addSchedule(){
```

Monthly Calendar 의 날짜 출력 및 이전/다음 버튼을 눌렀을 때 달이 넘어가도록 구현

```
299      public void dayPrint(int y, int m){
```

- WeekCalendar.java

Weekly Calendar 의 날짜 출력 및 이전/다음 버튼을 눌렀을 때 달이 넘어가도록 구현

```
219      private void updateWeek() {
```

```
225      public void weekPrint(int y, int m, int d) {
```

나머지 버튼 기능 구현은 위의 Family Calendar 와 유사

- DayCalendar.java

Daily Calendar 의 날짜 출력 및 이전/다음 버튼을 눌렀을 때 날짜가 넘어가도록 구현

```
216      private void updateDate() {
```

```
222      public void dayPrint(int y, int m, int d) {
```

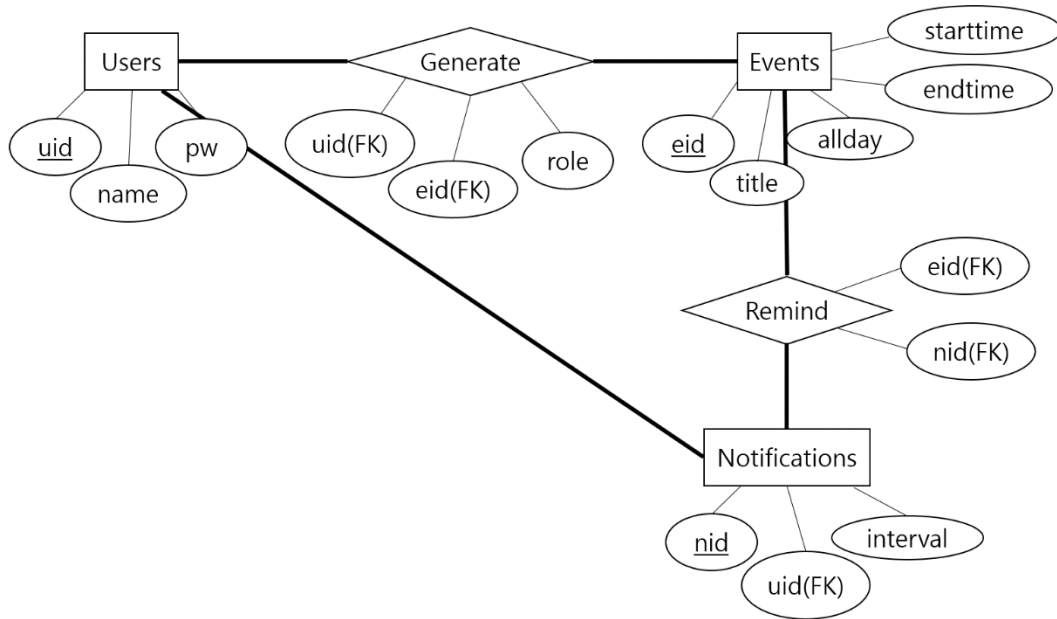
시간대 입력을 위한 구현

```
83          timeLabel[0] = new JLabel(text:"<html><body><br></body></html>");
84          leftPanel.add(timeLabel[0]);
85          for (int i=1; i<26; i++){
86              timeLabel[i] = new JLabel(i-1+" :00");
87              leftPanel.add(timeLabel[i]);
88          }
```

나머지 버튼 기능 구현은 위의 Family Calendar 와 유사

스키마 구현

스키마 구현을 위해, 아래와 같이 BCNF 를 만족하는 간단한 ER 다이어그램을 구상하였다.



- Users (uid, name, pw)

→ 모두 varchar / uid 는 이메일 입력

```
yschoi=# create table Users
yschoi-# (uid varchar(100) not null primary key,
yschoi-# name varchar(50) not null,
yschoi-# pw varchar(100) not null
yschoi-# );
```

- Events (eid, title starttime, end, allday)

→ eid & title: varchar / starttime & end: timestamp / allday: boolean

```
yschoi=# create table Events
yschoi-# (
yschoi-# Eid varchar(5) not null primary key,
yschoi-# title varchar(100) not null,
yschoi-# starttime timestamp not null,
yschoi-# endtime timestamp not null,
yschoi-# allday boolean not null
yschoi-# );
```

- Generate (uid(FK), eid(FK), role)

→ 모두 varchar / role 의 value 로는 'host'와 'attendee'만 입력

```
yschoi=# create table Generate
yschoi-# (
yschoi-# Uid varchar(100) not null,
yschoi-# Eid varchar(5) not null,
yschoi-# Role varchar(10) not null,
yschoi-# foreign key (Uid) references Users(Uid),
yschoi-# foreign key (Eid) references Events(Eid)
yschoi-# );
```

- Notifications (nid, uid(FK), interval)

→ nid & uid: varchar / interval: integer(몇 시간 전에 알림 공지를 띄울 것인지 입력)

```
yschoi=# create table Notifications
yschoi=# (
yschoi(# Nid varchar(5) not null primary key,
yschoi(# Uid varchar(100) not null,
yschoi(# interval integer not null,
yschoi(# foreign key (Uid) references Users(Uid)
yschoi(# );
CREATE TABLE
```

- Remind (eid(FK), nid(FK))

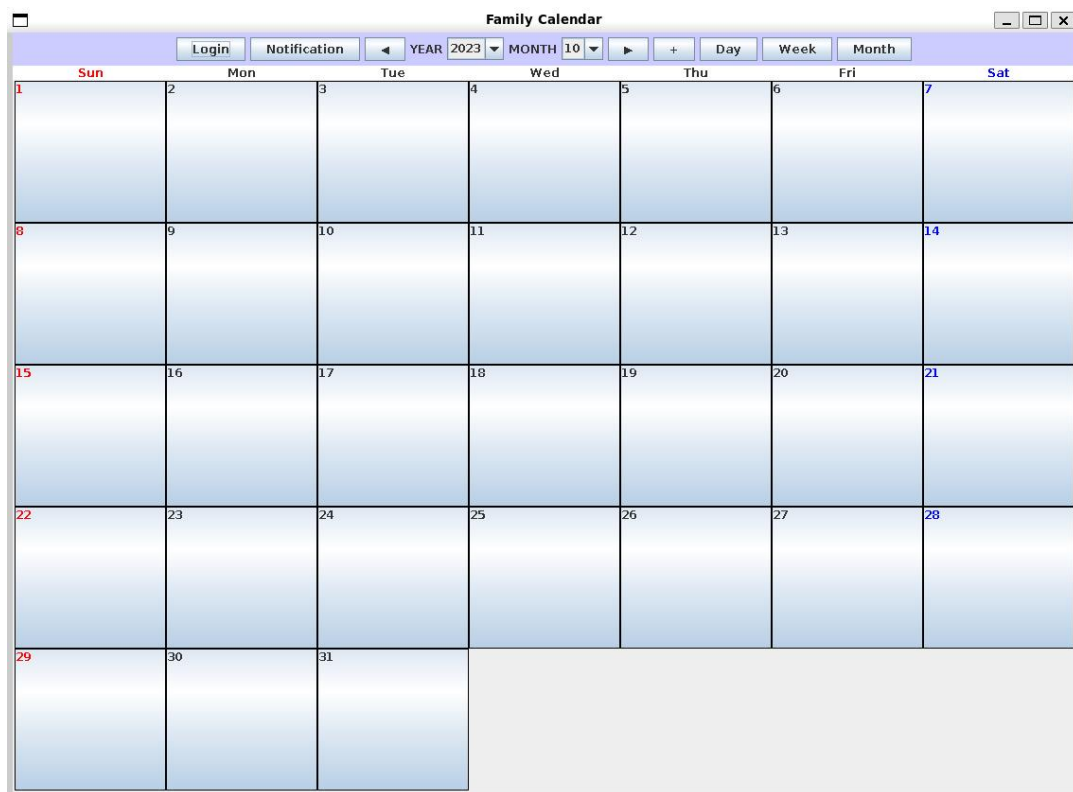
→ 모두 varchar

```
yschoi=# create table Remind
yschoi=# (
yschoi(# Eid varchar(5) not null,
yschoi(# Nid varchar(5) not null,
yschoi(# foreign key (Eid) references Events(Eid),
yschoi(# foreign key (Nid) references Notifications(Nid)
yschoi(# );
```

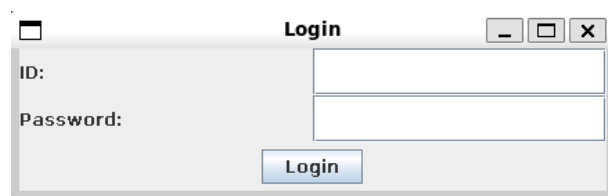
✓ Testing

· Java UI 및 기능

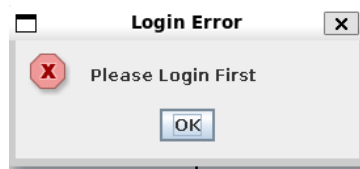
- Family Calendar 를 실행시켰을 때 나오는 첫 화면이다.



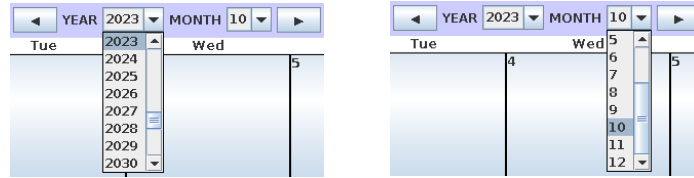
- 'Login' 버튼을 누르면 다음 UI 가 나온다. (테스트 로그인을 위해서 ID 와 PW 모두 '1234'로 설정해 두었다.)



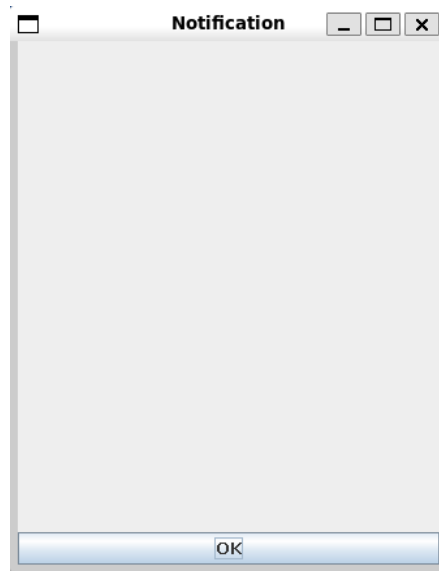
- 로그인을 하지 않으면, 날짜 변경 이외에는 어떠한 기능도 작동되지 않으며, 다음과 같은 에러가 뜬다.



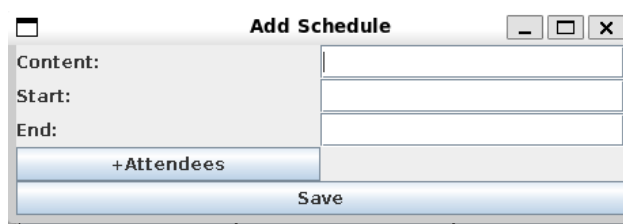
- 다음의 콤보 박스와 이전/다음 버튼을 눌러 달을 바꿀 수 있다.



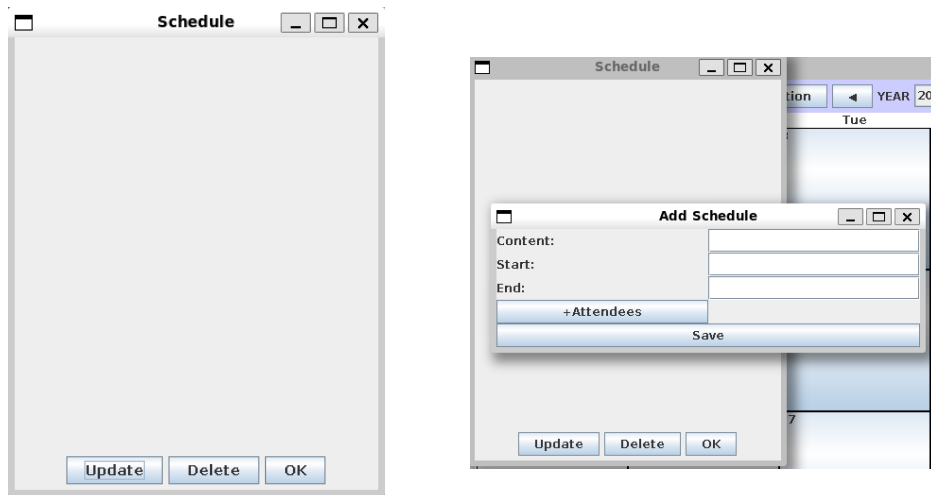
- Notification 버튼을 누르면 해당 달의 일정에 대한 알림이 뜬다. 이 창에 뜬 알림들은 해당 일정이 시작되기 직전까지 보이다가, 당 일정이 시작되면 자동으로 삭제되도록 API 를 구현할 예정이다.



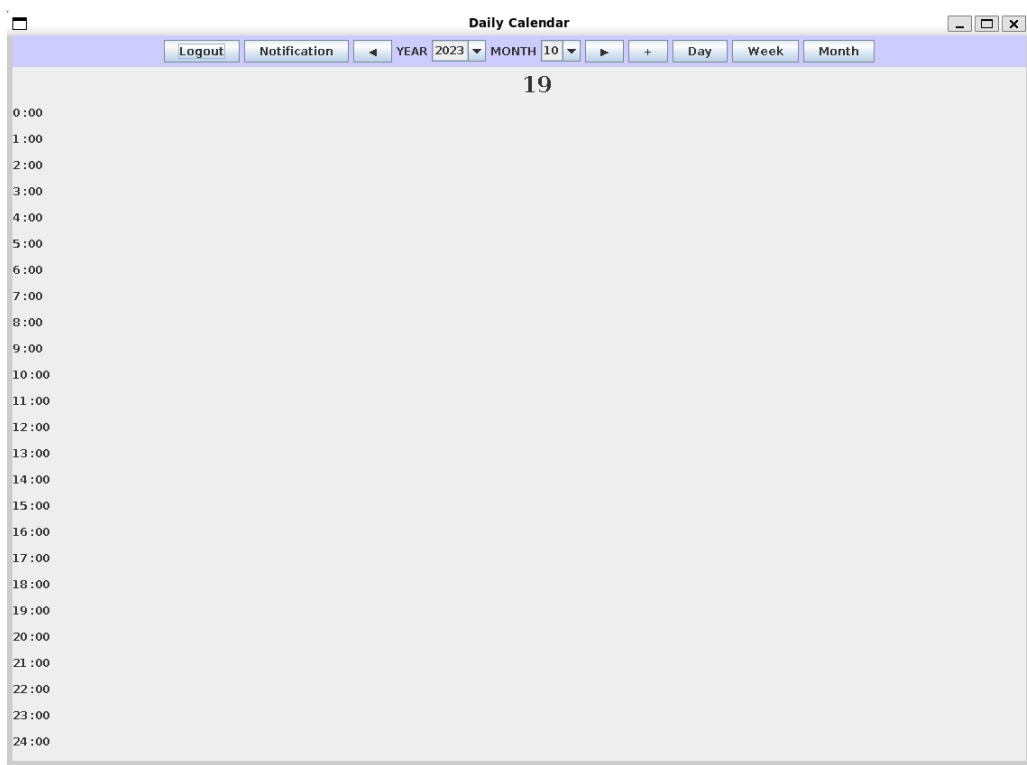
- +버튼을 누르면 일정을 추가할 수 있다. 해당 일정의 제목(Content), 시작 시점, 끝나는 시점을 입력할 수 있고, 일정에 참가하는 사람들을 +Attendees 버튼을 통해 초대할 수 있다.



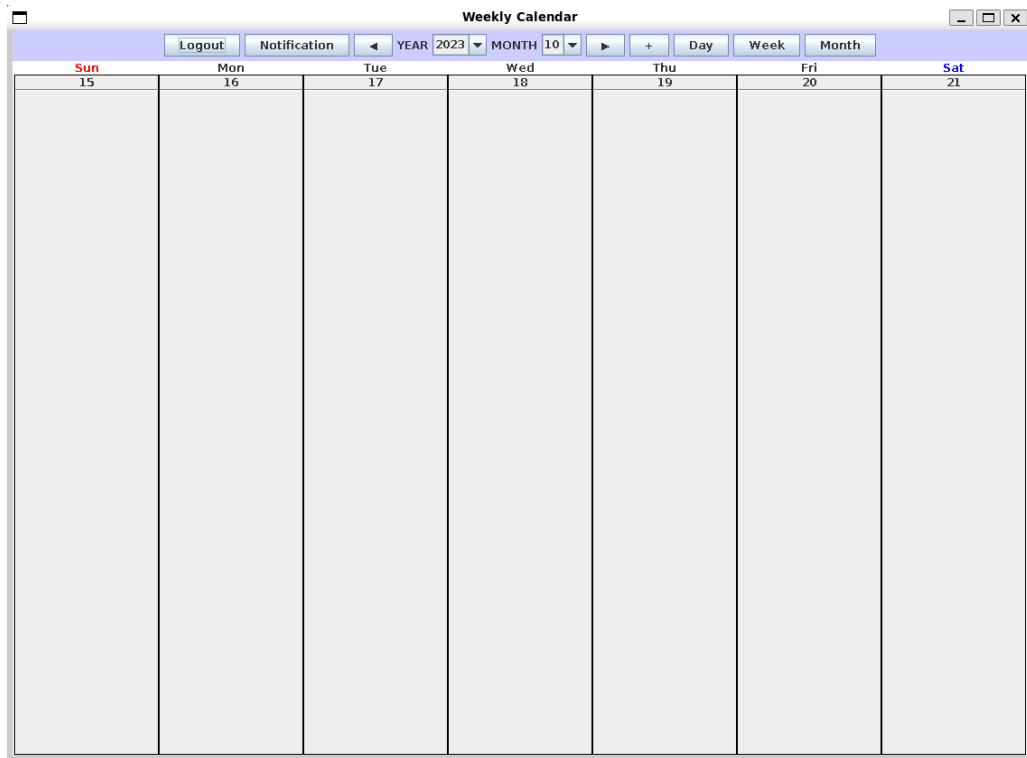
- 로그인 한 후 각 날짜의 버튼을 누르면 해당 날짜의 일정을 새로운 창으로 확인할 수 있다. 또한, 그 창에서 Update 버튼을 누르면 다시 일정 추가 창이 뜬다. Delete 버튼을 누르면 해당 일정이 삭제된다.



- Day 버튼을 누르면 Daily Calendar 화면으로 넘어간다. 이전/다음 버튼을 누르면 이전/다음 날짜로 넘어간다.



- Week 버튼을 누르면 Weekly Calendar 화면으로 넘어간다. 이전/다음 버튼을 누르면 이전/다음 주로 넘어간다.



- 'Month' 버튼을 누르면 다시 Monthly Calendar 화면으로 넘어간다. 또한, 'Logout' 버튼을 누르면 달력에 표시된 모든 일정이 보이지 않으며, 초기 화면(Monthly Calendar)으로 돌아간다.

스키마 구현

만들어진 Table 들 확인

- Users

```
yschoi=# select * from Users;
uid | name | pw
-----+-----+----
(0 rows)
```

- Events

```
yschoi=# select * from Events;
eid | title | starttime | endtime | allday
-----+-----+-----+-----+-----
(0 rows)
```

- Generate

```
yschoi=# select * from Generate;
uid | eid | role
-----+-----+----
(0 rows)
```

- Notifications

```
yschoi=# select * from Notifications;
nid | uid | interval
-----+-----+-----
(0 rows)
```

- Remind

```
yschoi=# select * from Remind;
eid | nid
-----+-----
(0 rows)
```

✓ Known problems

아직 Java UI 와 DB 를 연결하지 않았기 때문에 UI 의 완벽한 구현은 되지 않는다. 따라서 현재까지의 구현 중 발생하는 문제 및 다음 과제를 통해 해결할 문제에 대해 서술하고자 한다. 대부분의 문제는 API 와 JDBC 와 관련이 있다.

1. Monthly Calendar 에서 일정 확인을 위해 날짜 버튼을 클릭했을 때, 일정 확인창과 해당 날짜 데이터는 아직 연동되지 않는다.
2. 특정 이벤트에 참가자를 초대할 때, 상대방에게 초대 공지가 가도록 해야 한다.
3. Monthly, Weekly, Daily Calendar 중 어느 하나의 Calendar 에 일정이 입력되면, 그 일정에 맞게 나머지 Calendar 에도 똑같이 일정이 입력되어야 한다. 예를 들어, 2023.10.20 에 일정을 추가했다면, Weekly Calendar 와 Daily Calendar 의 10 월 20 일에 같은 일정이 시간대에 맞춰 입력될 수 있어야 한다.
4. 일정 추가에서 start time 과 end time 을 작성하면, 그 text 가 DB 에 timestamp 자료형으로 입력될 수 있어야 하고, 그에 맞춰 Calendar UI 에도 표기되어야 한다.

✓ Comments

- 과제에 얼마나 많은 시간을 할애했는가?
 - 보름정도 기간동안 과제를 수행하였고, Java UI에 70%, 스키마 구현에 30% 정도의 시간을 할애했던 것 같다.
- 설계, 코드 작성, 테스트는 어떻게 분배했는가?
 - 설계를 위해서 직접 손으로 UI와 ER 다이어그램을 그리며 구상하였다. 손으로 그린 UI를 바탕으로 Java 코드를 구현하였고, Java 파일 간 연동을 위하여 class 상속이 되도록 하였다. 또한, ER 다이어그램을 바탕으로 정규화를 자연스럽게 해결하였고, PK, FK를 지정하여 테이블을 만들었다.
- 과제에서 어려웠던 점은 무엇인가?
 - Java로 프로젝트를 해보았던 경험이 없어 익숙하지 않아 UI를 구현하는데 많은 시간을 들였다.
 - 스키마를 짤 때, Calendar에 적용되어야 할 기능을 모두 구현하기 위해 entity간 relation을 고민하는데 많은 시간을 할애했었다.
- 과제에 대해 바꾸고 싶은 점이 있는가?
 - Requirements를 현재보다 줄이고, 좀 더 구체적이었으면 좋겠다.