

세 번째 과제 : 캘린더 API 설계 및 DB연동 & 응용 전체 통합 테스트 및 검증








2018009125 조성우

1. Summary

과제 설명에 앞서 몇 가지 알려드릴 사항이 있습니다. 대부분의 구현을 끝마쳤지만 구현을 하지 못한 몇 가지 요소가 있습니다. 알람 기능, 주/일 단위로 캘린더 전환, 이벤트 overlap check 크게 이 세가지 부분을 구현하지 못했습니다. 알람 기능, 주/일 단위 캘린더 전환은 제 능력이 부족하여 적절한 구현법을 생각하지 못하여 구현하지 못했지만 이벤트 overlap은 어느정도 허용이 가능하다고 생각했습니다. 예를 들어, 약속1이 있고 약속1 중간에 약속2가 시작되는데 사용자 입장에서 약속1 도중에 나와서 약속2에 참여하는 상황이 가능하기도 하고, 약속2에 참석하지 못하더라도 어떤 일정이 있는지 확인만 하는 용도로 약속2를 캘린더에 남길 수도 있기 때문입니다. 또한 event 유효기간이 지나도 삭제하지 않았습니다. 실제 calendar에서도 삭제하지 않을 뿐 더러 일정을 기록으로 남겨둘 필요도 있을 것 같아 그대로 남겨두도록 하였습니다. (감점 사유라면 달게 받겠습니다..) 그리고 UI, 스키마도 많이 변경되었습니다. API를 만드는 과정에서 전에 과제로 제출했던 UI의 대부분이 변경되었으며 스키마도 변경되었으므로 확인 부탁드립니다.

























2. Specification/implementation/test

저는 리눅스에서 실행하는 대신(리눅스가 오류로 인해 실행되지 않았습니다..) 윈도우에서 postgresql-42.7.0버전으로 pgadmin4라는 프로그램을 사용해 DB를 연동하여 관리하였고 이 DB와의 연결은 실습시간에 했던 connection 방법대로 진행하였습니다. 즉 조교님이 공지사항에 올려 주신 스크린샷과 똑같이 진행하였습니다... code부분에 대한 설명은 주석으로 대체하도록 하고 specification에서는 실제 calendar가 작동하는 과정을 큰 그림에서 설명하도록 하겠습니다. 먼저 스키마부터 설명하도록 하겠습니다. 총 4개의 table이 존재합니다. 먼저 user들의 정보를 저장하는 users table입니다.





	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
 	id	character varying v	10		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 	password	character varying v	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 	uid	integer v			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	nextval('us'
 	position	character varying v	20		<input type="checkbox"/>	<input type="checkbox"/>	
 	name	character varying v	20		<input type="checkbox"/>	<input type="checkbox"/>	

Position은 가족에서의 직위입니다. (엄마,아빠,아들 등등) id와 password를 통해 로그인할 수 있습니다. uid가 primary key이며 serial type으로 설정하였습니다.

다음은 event들의 정보를 저장하는 events table입니다.













	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
 	eventid	integer v			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	nextval('ev
 	year	integer v			<input type="checkbox"/>	<input type="checkbox"/>	
 	month	integer v			<input type="checkbox"/>	<input type="checkbox"/>	
 	day	integer v			<input type="checkbox"/>	<input type="checkbox"/>	
 	hour	integer v			<input type="checkbox"/>	<input type="checkbox"/>	
 	min	integer v			<input type="checkbox"/>	<input type="checkbox"/>	
 	location	character varying v	30		<input type="checkbox"/>	<input type="checkbox"/>	
 	content	character varying v	40		<input type="checkbox"/>	<input type="checkbox"/>	
 	duration	integer v			<input type="checkbox"/>	<input type="checkbox"/>	
 	interval	integer v			<input type="checkbox"/>	<input type="checkbox"/>	
 	timeframe	integer v			<input type="checkbox"/>	<input type="checkbox"/>	
 	owner	integer v			<input type="checkbox"/>	<input type="checkbox"/>	

Eventid가 primary key입니다. Owner를 통해 이 이벤트를 생성한 사람의 uid를 함께 저장하였습니다. 다음은 어떤 user가 어떤 event를 가지고 있는지에 대한 정보가 저장되어 있는 user_event table입니다.

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
 	user_id	integer v			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
 	event_id	integer v			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

User_id와 event_id 한 쌍이 primary key입니다. 마지막으로 rsvp에 대한 정보들을 저장하고 있는

table인 rsvps table입니다.

Columns							
	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
 	hid	integer v			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
 	gid	integer v			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
 	r_content	character varying v	30		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
 	r_hour	integer v			<input type="checkbox"/>	<input type="checkbox"/>	
 	r_min	integer v			<input type="checkbox"/>	<input type="checkbox"/>	
 	r_check	integer v			<input type="checkbox"/>	<input type="checkbox"/>	

hid(host uid), gid(guest uid), r_content(rsvp의 내용) 전체 쌍이 primarykey입니다. r_check(0: 응답 전 , 1: 수락, 2: 거절)를 통해 응답여부/수락여부를 확인합니다.

이제 UI와 함께 전체적으로 설명하도록 하겠습니다.

로그인 창

아이디

비밀번호

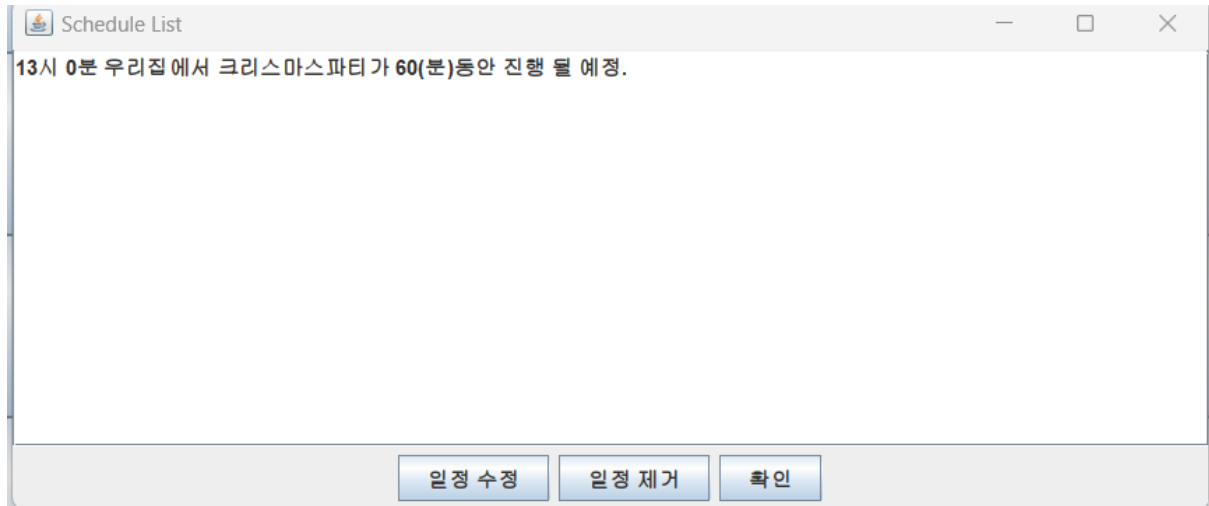
회원가입

로그인

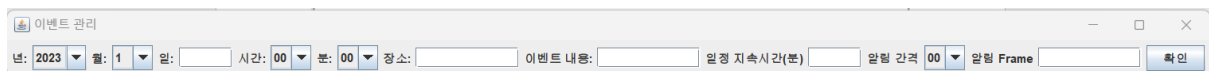
SwingLogin을 실행했을 때 화면입니다. Calendar 프로그램은 SwingLogin을 실행 함으로써 시작됩니다. 즉, 실제 프로그램의 실행은 SwingLogin을 실행 함으로써 시작되는 것이므로 전체 흐름을 확인하시려면 SwingLogin부터 실행시켜서 확인하시면 됩니다. 아이디와 비밀번호를 입력받도록 하였고 회원가입도 가능하도록 하였습니다. 다음은 회원가입 UI입니다.

회원가입버튼을 누르면 위와 같은 UI창이 뜨며 입력된 text들은 id, password, name, position로 각각 users table에 저장됩니다. 다시 로그인 창으로 돌아와서 로그인에 실패하면 실패 문구가 뜨고 성공하면 다음과 같은 창이 뜨게 됩니다.

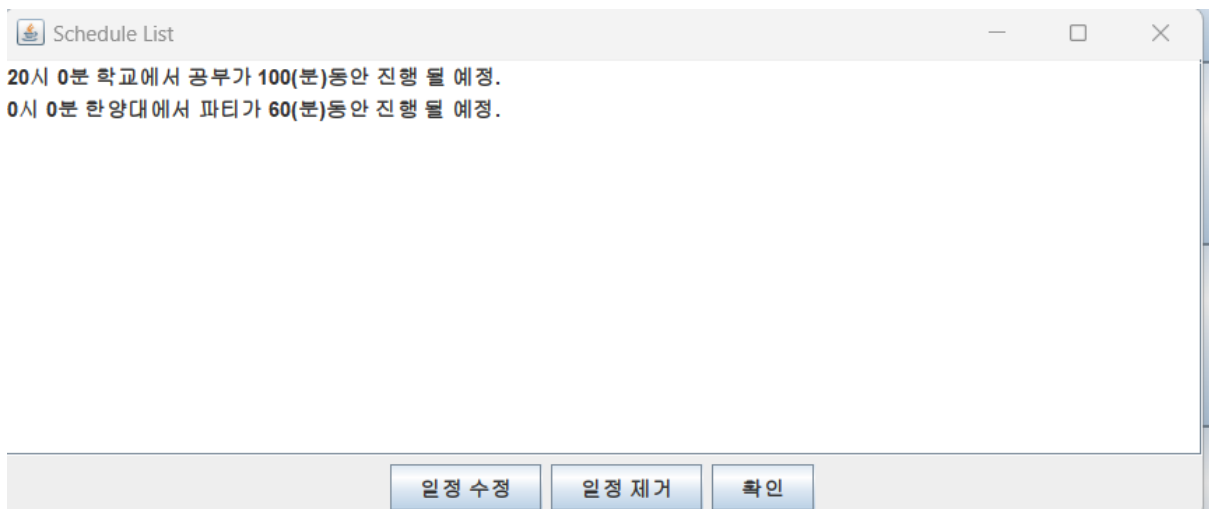
현재 월에 맞춰 자동으로 해당 월의 calendar가 뜨도록 하였으며 calendar가 등장함과 동시에 uid를 파악하여 해당 uid가 가지고 있는 event들이 자동으로 보이도록 하였습니다. 왼쪽 배너에는 RSVP요청과 관련된 기능들이 있고 바닥에는 일정 추가/검색, 오른쪽에는 가족들의 availability를 확인하는 버튼과 회원정보 수정 버튼 그리고 전체 가족 목록 확인 버튼이 있습니다. 사진 예시를 보면 25일에 미리 크리스마스 파티라는 일정이 추가가 되어있는데 25일을 클릭하게 되면



다음과 같이 event의 세부사항이 등장합니다. 일정을 수정해보겠습니다.

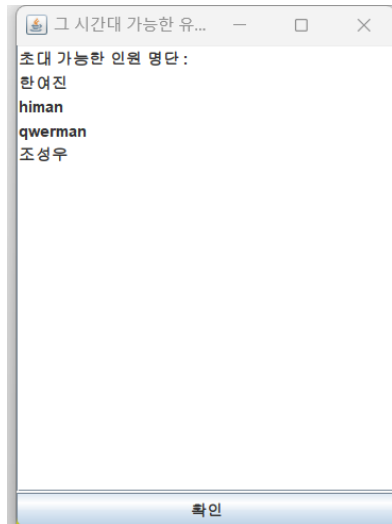


다음과 같은 UI가 등장하며 일정의 세부사항을 변경할 수 있도록 합니다. 알람 기능을 구현하지는 못했지만 알람의 간격과 frame까지 받는 것을 알 수 있으며 참고로 일정 지속시간은 항상 분으로 받도록 하였습니다. 27일로 날짜를 바꾸고 그날 학교에서 공부한다는 내용을 입력해 보았습니다. 그리고 27일을 클릭해보면



다음과 같이 원래 27일에 있던 일정인 파티와 함께 이벤트 세부내역이 나오는 것을 알 수 있습니다. 삭제하면 사라지게 되고 events table에서 사라지게 됩니다. 이렇게 이벤트의 수정/삭제/추가 가 있을 때마다 CalendarMain안에 있는 Update함수를 통해 계속 calendar UI를 update합니다. 모든 기능을 하나하나 리뷰하면 양이 너무 많아질 것 같아서 중요한 내용들 위주로 설명하겠습니다. Calendar main 화면 밑에 있던 추가 기능도 사실 수정 기능 UI랑 같습니다. 다른 점은 일정 추가 버튼을 눌렀을 때 유저들을 추가하는 UI가 표시된다는 점입니다. 일정 검색기능은 장소, 이벤트 내용을 입력으로 받는 칸이 있는데 장소에 예를 들어 한양대라고 치면 장소가 한양대인 event가

검색됩니다. 이벤트 내용도 이와 마찬가지로 빈칸으로 두면 그 부문에서는 필터를 적용하지 않는다는 뜻입니다. 그리고 시작날짜와 종료날짜도 입력 받는데 이 두개는 필수이며 시작날짜와 종료날짜 사이에 진행되는 event를 검색하게 됩니다. Main calendar 오른쪽 위에 특정 시간 가능 인원 확인 버튼을 누르고 날짜와 시간을 입력하면



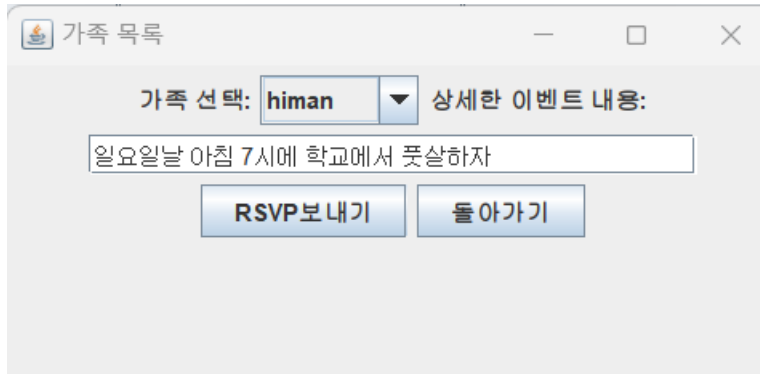
다음과 같이 해당 날짜와 시간에 일정이 없는 가족목록이 UI에 표시됩니다. 다음은 회원 목록 수정 버튼을 누르면 다음과 같은 UI가 표시됩니다.



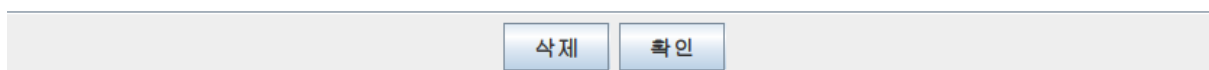
현재 아이디:chosw05222 ->	<input type="text"/>
현재 비밀번호:@playjoy2 ->	<input type="text"/>
현재 이름:조성우 ->	<input type="text"/>
현재 역할:아들 ->	<input type="text"/>
수정하기	돌아가기

정보를 수정할 수 있으며 입력된 text내용으로 DB를 Update합니다. 전체 가족 목록 확인 버튼은 단순히 모든 유저(가족)의 이름을 표시합니다. 이제 RSVP에 관련된 내용을 설명하겠습니다.

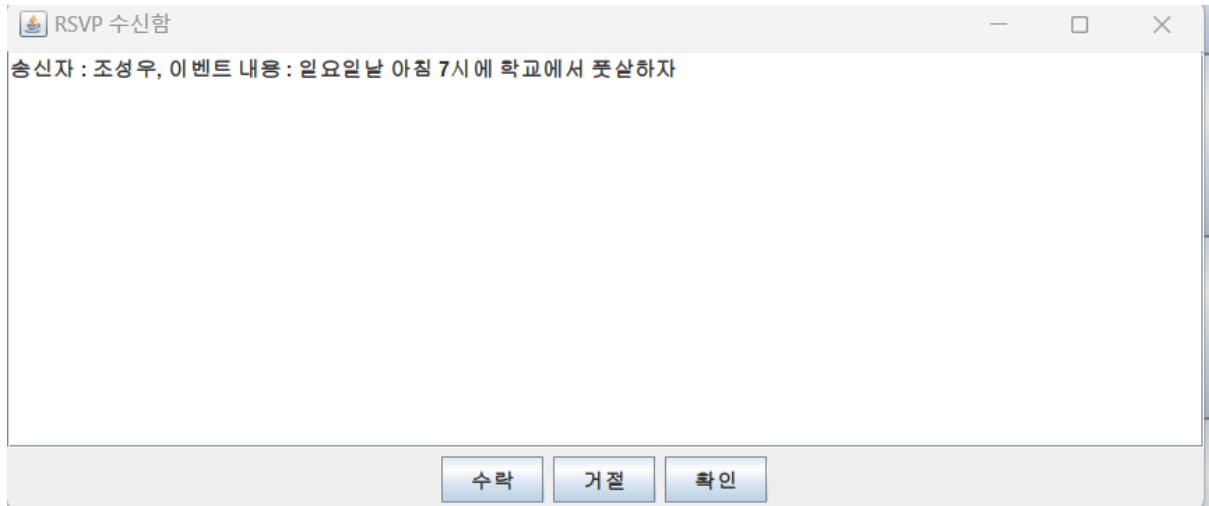
우선 RSVP 보내기버튼을 눌러 다음과 같이 himan에게 RSVP를 보내 보겠습니다.



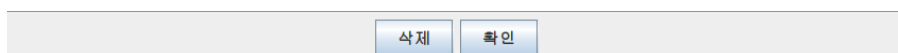
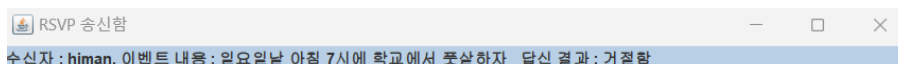
이렇게 가족을 선택하고 일정이 되는지를 확인하기 위해 다음과 같은 문구를 보내 보겠습니다. 참고로 가족선택을 누르면 현재 모든 가족의 이름이 표시되고 한 번에 한명에게만 보낼 수 있습니다. (여러명에게 RSVP보내려면 여러 번 보내기 버튼을 클릭해 보내야 합니다.) 이렇게 보내면 rsvps table에 insert가 됩니다. 이번엔 RSVP 송신함에 들어가보겠습니다.



다음과 같이 방금 내가 보낸 RSVP request가 들어있는 것을 확인할 수 있습니다. 물론 확인전에 삭제할 수도 있습니다. 그럼 himan 계정으로 들어가 수신함을 살펴보겠습니다.



수신함에 다음과 같이 수락/거절 버튼이 있습니다. 수락을 누르면 DB에서 update를 통해 r_check가 1로 설정되고 거절을 하면 r_check가 2로 설정됩니다. 10분이 지나면 자동으로 거절처리를 하였습니다. r_check는 -0이지만 host쪽에서 송신함을 확인할 때 10분이 지났으면 거절함으로 표시하도록 하였고 수신자 입장에서도 10분이 지난 request는 수신함에 표시되지 않도록 하였습니다. 다음은 실제로 10분이 지난 후 host측에서 송신함을 확인했을 때의 결과입니다.



guest측에서는 수신함에 아무것도 없습니다. 따로 스크린 샷을 첨부하진 않겠습니다. 이 정도면 전체적으로 큰 그림에서 calendar에 관해 설명한 것 같습니다. 그리고 rsvp요청은 송신자가 송신함에서 해당 rsvp를 삭제해야지 DB에서 삭제됩니다. Code부분의 내용을 설명하기에 너무나도 방대해서 여기다 설명하지 못하였습니다. 따라서 Code부분에 간략한 주석들을 첨부하였으니 참고 해주시면 감사드리겠습니다.

3. Comments

제가 제 구현을 pgAdmin4 프로그램을 통해서 DB랑 연결되는 것을 확인하고 그렇게 해서 과제를 진행하였는데 조교님 환경에서는 schema가 완벽하게 동일하지 않을 수도 있고 해서 DB랑 연동이 되지 않아 프로그램이 작동되지 않는 상황이 벌어지는 것이 걱정이 됩니다. 계정명과 비밀번호도 우선 dbms_practice로 바꾸고 스키마도 따로 첨부하도록 하겠습니다. 그래도 만약 DB연동이

되지 않아 프로그램이 제대로 굴러가지 않는다면... 저에게 메일 보내주시면 제가 찾아가서 테스트되는 것을 보여드리도록 하겠습니다. 그런 상황이 벌어지지 않기를 바라지만 만약 그렇게 된다면 불편을 드릴 수도 있을 것 같아 미리 죄송합니다.

제가 평소에 java를 다루지 않기도 하고 DB, UI쪽은은 아예 처음이어서 code를 보시면 많이 힘들 수도 있습니다. (특히 insert, select, update, select class code들이 처참한 것 같습니다. 죄송합니다.) 하지만, 이것저것 구글링을 해보며 어떻게든 완성시키려 노력하였고 UI가 조잡하긴 하지만 어느정도는 과제 명세에 적힌 부분들이 돌아가는 것 같습니다. 아쉬운 점은 알람 쪽을 건들지 못한 것입니다. (주/일 단위로 mainCalendar view를 바꾸는 부분은 UI쪽 인 것 같아 건들질 못하겠습니다...) 알람을 담당한 class가 background에서 계속 실행되어서 알람 시간이 될 때마다 알람을 보내야 할지, 따로 java 라이브러리에서 외부class를 가져와서 써야 할지 이런 것들이 정리되지 않아서 구현하지 못한 것 같습니다. 긴 글 읽으시느라 고생하셨습니다. 감사합니다.