

오픈소스 16조

동적 스케줄러 보고서

목차

I. 동적 스케줄러 화면

1. 달력 화면
2. 셀레니움 연동 화면
3. 메모 화면

II. 계획서 내용과의 비교

1. 수행한 부분
2. 수행하지 못한 부분
3. 변경된 내용
4. 소스코드 모듈 정리

III. 구현시의 어려움 및 해결

1. <16011158 이영호>
2. <17011651 이승우>
3. <17011646 이정호>

IV. 개발 툴 정리

1. JAVA
2. SEIENIUUM
3. CALENDAR FX
4. SQLITE
5. SCENE BUILDER

V. 결론

1. GITHUB 주소 및 기여도

I. 동적 스케줄러 화면

1. 달력 화면

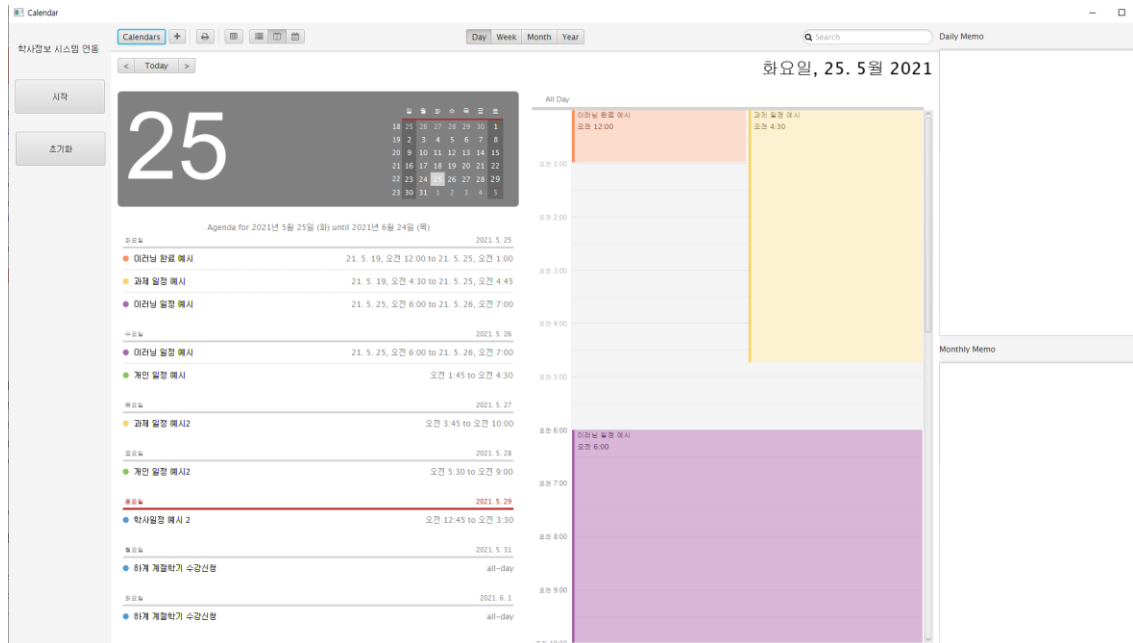
1. 달력 화면

2. 셀레니움 연동 화면

3. 메모 화면

1-1) 화면 구성

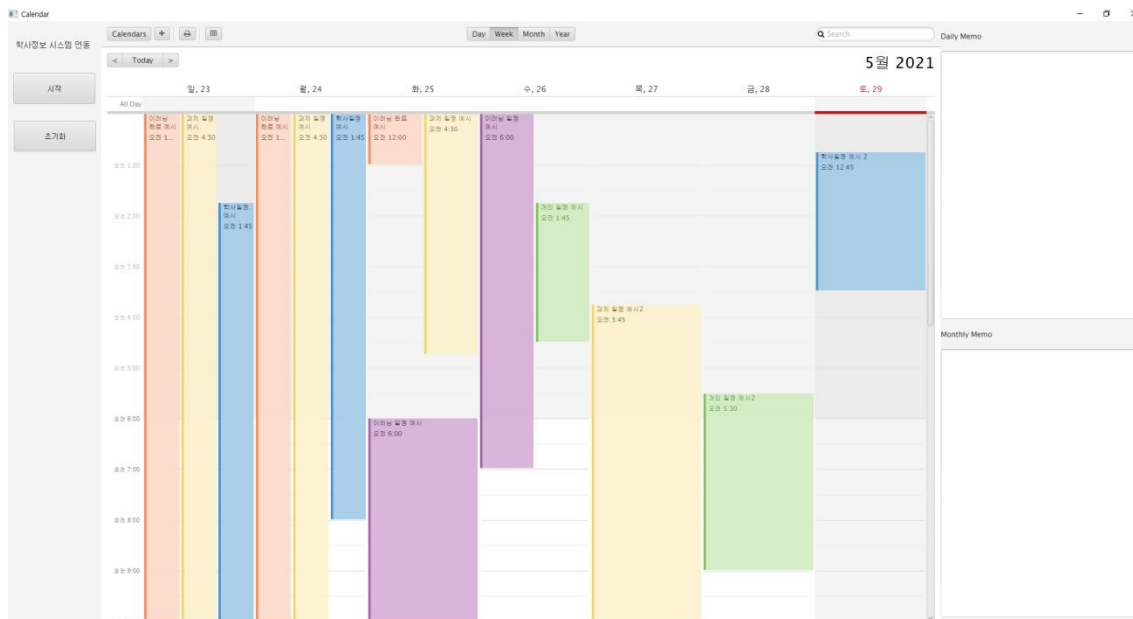
1) Day view



프로그램 실행 시, 처음으로 보게 되는 화면.

오늘을 기준으로 이후의 스케줄 목록 및 오늘의 타임테이블을 확인할 수 있다.

2) Week view



주 단위의 스케줄을 타임테이블의 형식으로 확인할 수 있다.

I. 동적 스케줄러 화면

1. 달력 화면

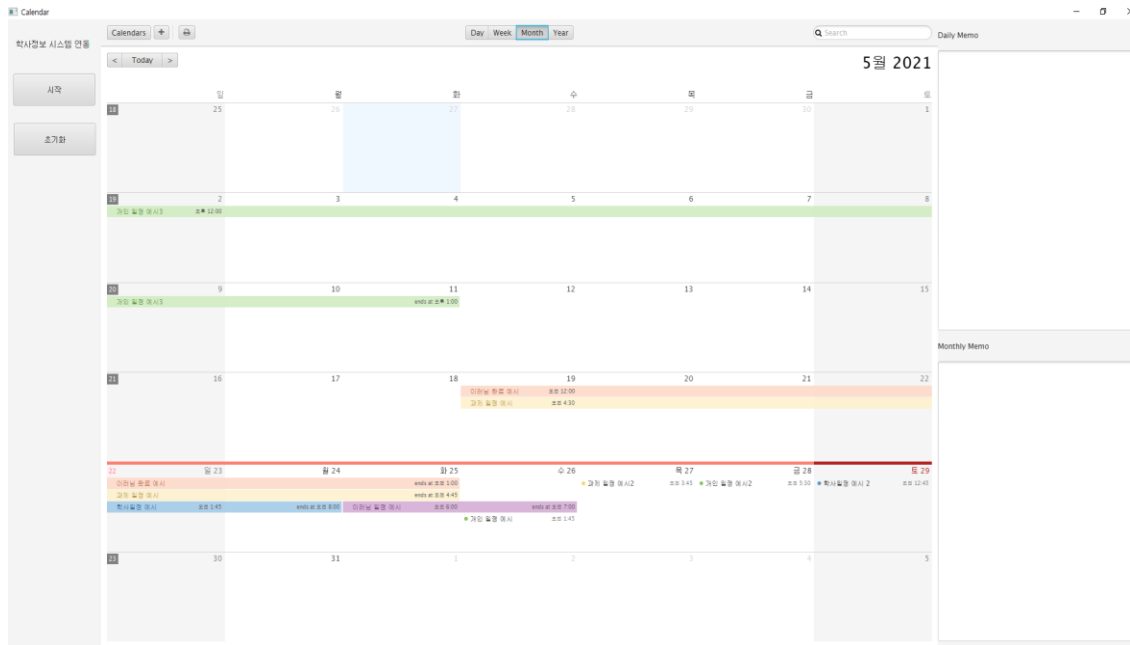
1. 달력 화면

2. 셀레니움 연동 화면

3. 메모 화면

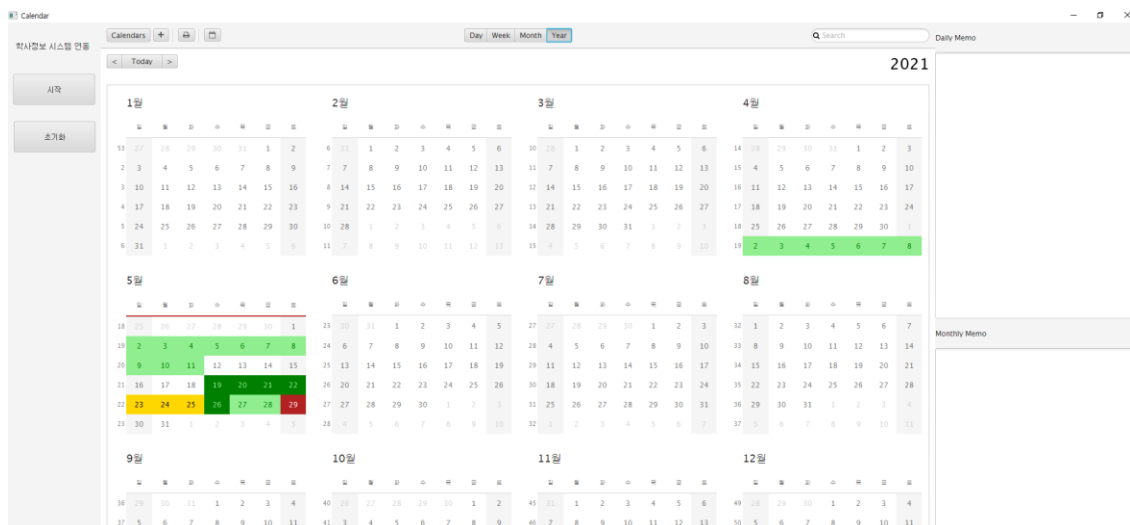
1-1) 화면 구성

3) Month view



월 단위의 스케줄을 캘린더 형식으로 볼 수 있다.

4) Year view



연 단위의 스케줄을 캘린더 형식으로 볼 수 있다.

I. 동적 스케줄러 화면

1. 달력 화면

1. 달력 화면

2. 셀레니움 연동 화면

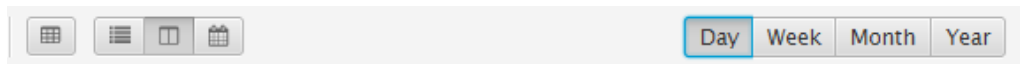
3. 메모 화면

1-2) 화면 조작법

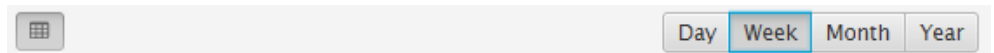


화면 상단의 Day, Week, Month, Year 버튼을 클릭 시, 각 화면으로의 전환이 가능하다.

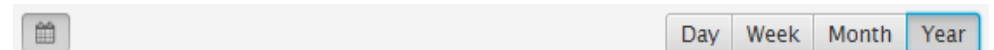
이때, Day, Week, Year화면의 경우 좌측에 존재하는 버튼을 이용하여 각 스케줄을 다른 양식으로 확인 할 수 있도록 해주는 기능을 제공한다.



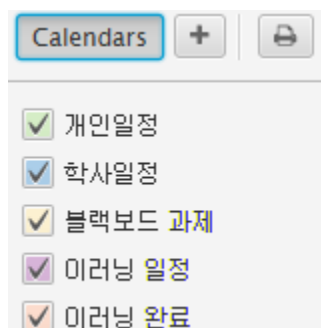
Day의 경우 좌측에 4개의 버튼이 존재하며, 일정의 종류가 구분되어진 타임테이블, 리스트, 리스트 및 타임테이블, 타임테이블 등의 양식으로 스케줄을 확인할 수 있다.



Week의 경우 좌측에 1개의 버튼이 존재하며, 일정의 종류가 구분되어진 타임테이블의 양식으로 주 단위의 스케줄을 확인할 수 있다.



Year의 경우 좌측에 1개의 버튼이 존재하며, 가로축은 월별, 세로축은 일별로 나누어진 스케줄을 확인할 수 있다.



Calendars 버튼 클릭시 보고싶은 정보를 선별하여 볼 수 있다. 우측의 프린트 버튼을 클릭시 필요한 스케줄을 인쇄할 수 있다.

1. 달력 화면
2. 셀레니움 연동 화면
3. 메모 화면

1.동적 스케줄러 화면

2.셀레니움 연동 화면

1-1) 연동 메뉴



화면 좌측에 위치하고 있다.

시작 버튼으로 블랙보드 정보를 연동할 수 있다.

초기화 버튼으로 개인 일정 외의 정보를 초기화 할 수 있다.

1-2) 연동 화면



시작 버튼 클릭 시 확인창이 뜨고 '예'를 클릭 시 크롬 블랙보드 로그인 화면이 생긴다.
사용자가 모든 정보를 '직접'입력 후 로그인하여야 정보 연동이 시작된다.

- 1. 달력 화면
- 2. 셀레니움 연동 화면
- 3. 메모 화면

I.동적 스케줄러 화면

3.메모화면

Daily Memo

일간메모를 작성해주세요

Monthly Memo

메모 창은 화면 우측에 위치하고 있으며, 화면의 전환과 관계없이 모든 화면에서 이용할 수 있다.

DailyMemo, MonthlyMemo로 나누어져 있으며, 해당 창에 텍스트를 작성하면 작성되어진 내용은 src\application폴더 내부에 존재하는 DailyMemo.txt, MonthlyMemo.txt 파일에 저장되어지게 된다.

1. 수행한 부분
2. 수행하지 못한 부분
3. 변경된 내용
4. 소스코드 모듈 정리

II. 계획서 내용과의 비교

1. 수행한 부분

❖ 프로그램 GUI 설계, 배치

동적 스케줄러에 포함되어야 할 GUI들의 설계나 배치를 해주고 이를 구현해주었다.

중앙부분에는 CalendarFX를 통한 달력부분을 구현, 왼쪽에는 학사정보시스템과 블랙보드를 연동해주는 시작버튼과 이를 초기화 해주는 종료버튼이 있다.

또한 Calendars 버튼을 통해 개인일정, 학사일정, 이러닝 일정 등 따로 확인 할 수 있는 버튼을 만들었다.

오른쪽에는 일간, 월간메모장이 있으며 정보가 유지 되도록 작성되었다.

❖ 블랙보드 내용 크롤링

셀레니움을 통한 블랙보드, 학사정보를 받아온다.

연동은 실시간이 아닌 사용자가 직접 연동버튼을 통해 원하는 시간에 연동을 할 수 있게 제작되었다.

❖ 데이터베이스 제작

SQLite를 통해 앱 내부에서 데이터를 저장할 수 있도록 만들어졌다. 셀레니움을 통해 만들어진 데이터들을 데이터베이스화 시켜주었다.

1. 수행한 부분
2. 수행하지 못한 부분
3. 변경된 내용
4. 소스코드 모듈 정리

II. 계획서 내용과의 비교

2. 수행하지 못한 부분

구현 일정의 예상과 구현 마감일자의 변동 때문에 프로그램 완성 후 예정이었던 앱 부분을 제외하고는 계획서 내의 중요부분들은 거의 구현을 다 한 것으로 판단되어 수행하지 못한 부분은 따로 없다.

1. 수행한 부분
2. 수행하지 못한 부분
3. 변경된 내용
4. 소스코드 모듈 정리

II. 계획서 내용과의 비교

3. 변경된 내용

❖ MATERIAL DESIGN 에서 CalendarFX 오픈소스 사용
처음 달력부분은 JavaFX와 Scenebuilder를 이용하여 자체제작을 하려고 계획하였다.

여기에 기초디자인을 담당하는 MATERIAL DESIGN를 추가하여 JavaFX 기초 컨트롤러를 보완하려 하였다.

하지만 이렇게 GUI를 전부 작성하기엔 제작기간이 매우 부족해져 달력을 이용할 수 있는 오픈소스를 찾아 사용하기로 하였다.

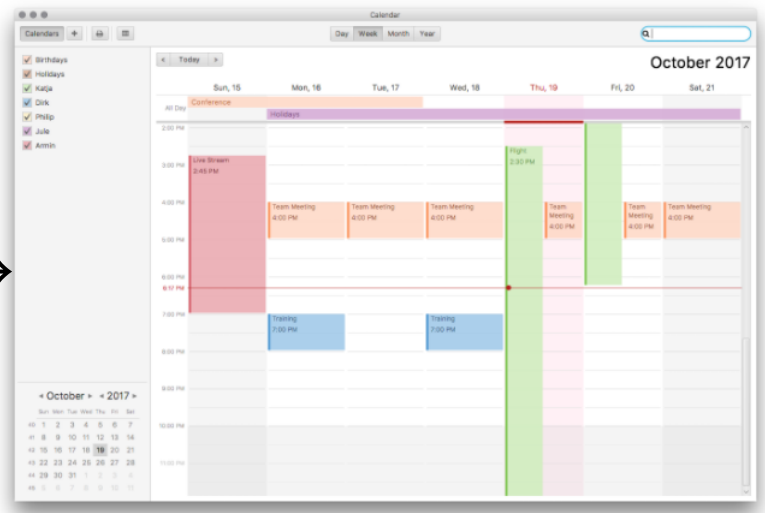
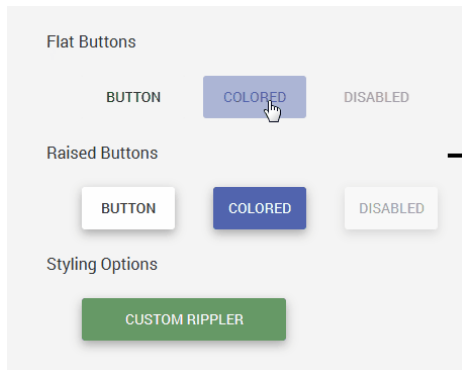
이는 작업시간을 단축시켜 주었으며 GUI의 디자인 부분에도 도움을 주었다.

CalenderFX Demo 실행화면

MATERIAL DESIGN인 오픈소스 Jfoenix의 버튼디자인

JFXButton

The JFXButton is a [material design compliant](#) implementation of



또한 초기의 화면구성인 메뉴, 타임테이블, 캘린더, 초기화면 에서 오픈소스가 제공해주는 View를 활용 하도록 계획을 변경하였다.

1. 수행한 부분
2. 수행하지 못한 부분
3. 변경된 내용
4. 소스코드 모듈 정리

II. 계획서 내용과의 비교

3. 변경된 내용

❖ MariaDB 에서 SQLite로 사용변경

계획에는 MariaDB를 이용하여 데이터를 저장하고 이를 이용하여 프로그램에 데이터를 저장, 활용을 할 계획이었다.

하지만 프로그램 배포 시에 해당 컴퓨터에 MariaDB가 설치되어 있어야 하고 이런 번거로움을 줄이고자 SQLite를 이용하여 프로그램의 내부에 데이터베이스를 만들어 활용할 수 있게 하였다.

SQLite를 이용한 학업데이터 구조

이름	타 스키마
<div> <div>데이터베이스 (5)</div> <div> <div>BlackBord_Finished_Schedule</div> <div>BlackBord_Movie_Schedule</div> <div>BlackBord_Schedule</div> <div>Private_Schedule</div> <div>University_Schedule</div> </div> </div>	<div> <div>CREATE TABLE "BlackBord_Finished_Schedule" ("Title" VARCHAR(255), "FullDay" INTEGER, "StartDate" Date, "EndDate" Date, "StartTime" Time, "EndTime" Time)</div> <div>CREATE TABLE "BlackBord_Movie_Schedule" ("Title" VARCHAR(255), "FullDay" INTEGER, "StartDate" Date, "EndDate" Date, "StartTime" Time, "EndTime" Time)</div> <div>CREATE TABLE "BlackBord_Schedule" ("Title" VARCHAR(255), "FullDay" INTEGER, "StartDate" Date, "EndDate" Date, "StartTime" Time, "EndTime" Time)</div> <div>CREATE TABLE "Private_Schedule" ("Title" VARCHAR(255), "FullDay" INTEGER, "StartDate" Date, "EndDate" Date, "StartTime" Time, "EndTime" Time)</div> <div>CREATE TABLE "University_Schedule" ("Title" VARCHAR(255), "FullDay" INTEGER, "StartDate" Date, "EndDate" Date, "StartTime" Time, "EndTime" Time)</div> </div>
인덱스 (0)	
뷰 (0)	
트리거 (0)	

1. 수행한 부분
2. 수행하지 못한 부분
3. 변경된 내용
4. 소스코드 모듈 정리

II. 계획서 내용과의 비교

4. 소스코드 모듈 정리

1. App_Controller.java

APP 전체의 응답을 받는 컨트롤러, UI 의 버튼 응답을 처리함.

2. Data_base.java

SQLITE 와 연결하여 데이터 저장, 실행 시
데이터 복원, 셀레니움 데이터 연동 등, 데이터 관련
내용을 담당

3. Event_handle.java

날짜 변경, 시간 변경 캘린더 삭제 등의 캘린더
이벤트를 받아 처리하는 클래스.

4. Main.java

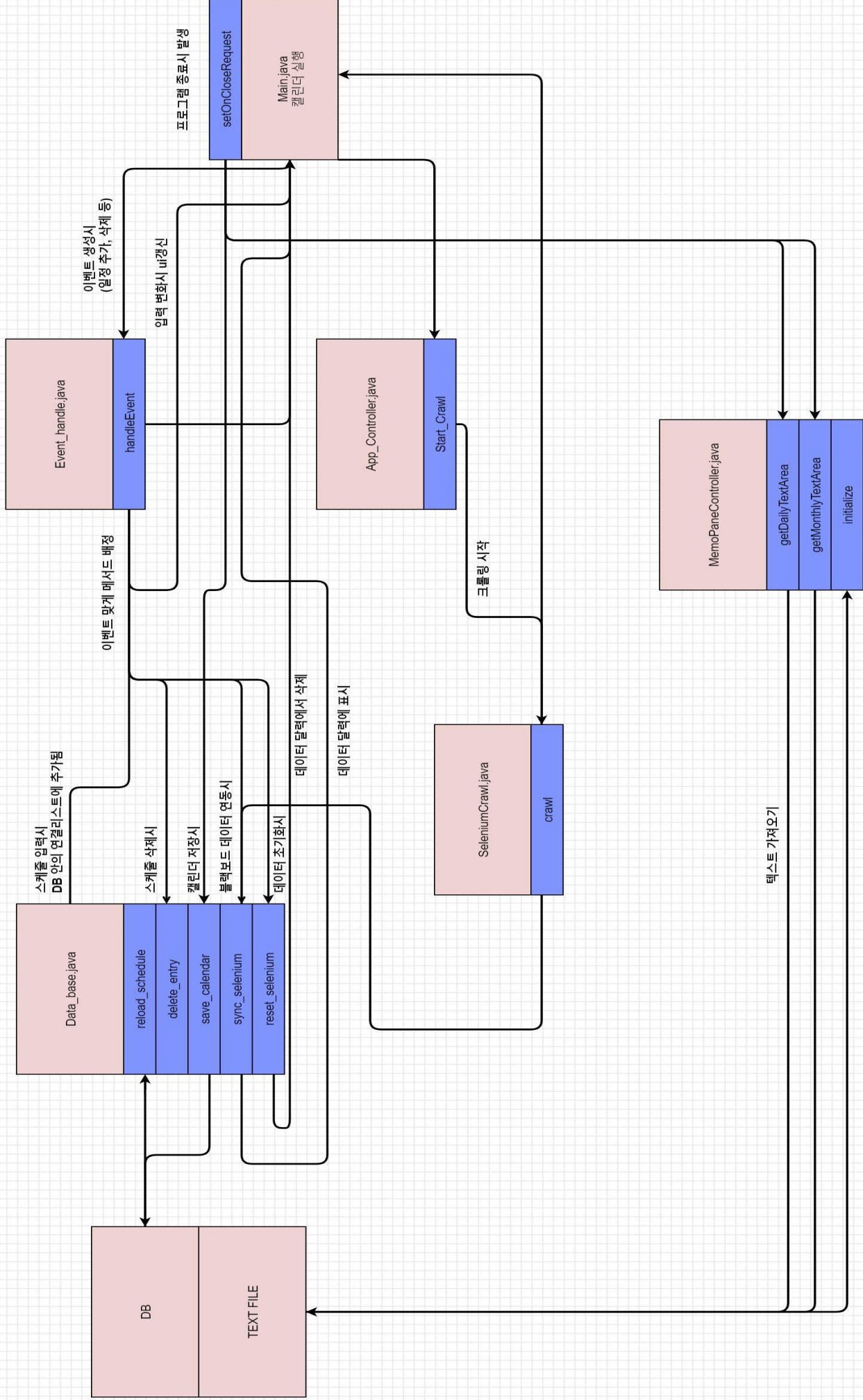
캘린더 메인 부분으로, 위의 Main 에서 캘린더 설정이
완료되고 캘린더가 실행됨

5. MemoPaneController.java

Scenebuilder로 작업해 나온 FXML파일에
이벤트 핸들러를 작성하여 저장.

6. SeleniumCrawl.java

블랙보드, 학사일정 데이터를 읽어와 데이터셋 클래스에
알맞은 형식으로 저장.



1. <16011158 이영호>
2. <17011651 이승우>
3. <17011646 이정호>

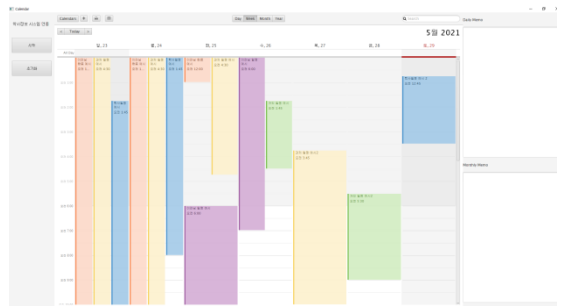
III. 구현시의 어려움 및 해결

<1601158 이영호>

처음에 UI에 대하여, 굉장히 쉽게 접근했었다. 특히
달력을 굉장히 쉽게 생각했었는데, 생각보다 구현 하는
과정에서 시간이 많이 걸린다는 것을 깨닫고, 이미
개발되어 있는 오픈소스를 찾게 되었다.

생각보다 달력 만드는 것이 까다로웠고 조금 괜찮은 라이브러리를 찾았다 싶으면, 다른 플랫폼이거나, 사용자가 쉽게 코드를 작성할 수 없고, 그대로 가져 써야 하는 라이브러리들이 상당히 많았다. 깃헙을 뒤진 끝에 Calendarfx 라는 오픈소스 라이브러리를 찾게 되었는데, Calendarfx 라이브러리도 친절한 라이브러리는 아니었지만, 열심히 알아본 결과 어떻게 쓰는 지 알아낼 수 있었다.

<처음부터 만든 것과 오픈소스 사용 비교>

[illegible]

실제로 화면 자체 구현은 왼쪽이 더 많이 걸렸지만
잘 만들어진 오픈소스를 이용한 오른쪽이 훨씬 더
보기 좋고, 사용하기에도 편했다. 단 Calendarfx 오픈
소스를 사용하려면, 오픈소스 자체를 다시 공부해야 했다.

1. <16011158 이영호>
2. <17011651 이승우>
3. <17011646 이정호>

III. 구현시의 어려움 및 해결

<16011158 이영호>

프로그램 실행되는 모습만 보면 Calendarfx가 상당히 준수한 UI를 제공해주는 덕분에, 그저 오픈소스만 가져다 썼다고 느낄 수도 있었다. 하지만 실제 Calendarfx를 실행 해 보면, UI 기능 외에 제공하는 것이 아무것도 없음을 느낄 수 있었다. 우선 데이터 저장부터 해결해야 되었다. 데이터 저장은 SQLITE 를 이용해서 캘린더 별로 테이블을 만들었지만, 캘린더를 저장할 때 계속해서 쿼리 문을 넣게 되면, 데이터베이스가 그 양을 감당하기 어려웠다. 따라서 프로그램 실행과 끝에만, 데이터를 저장하고 프로그램이 실행중인 동안에는 각 스케줄 객체들을 묶는 LINKED LIST를 구현하여, 변동사항들을 보관하고 있다가 프로그램이 종료 시 일괄적으로 쓰게 만들었다.

이렇게 구현하니 두 가지의 장점이 생겼다.

첫 번째는 위에 말한 데이터베이스 쿼리량의 수가 대폭 줄어들었다는 것이고, 두 번째는 사용자가 저장버튼을 누르는 등의 행위가 없어도, 달력이 종료될때 자동으로 저장이 되게 되었다.

그 뒤에는 , 달력들의 변동 사항을 어떻게 담아야 할 지가 중요했고, 구글링을 열심히 해 본 결과, 이벤트를 처리하는 Event handler 클래스를 만든다면, 어떤 스케줄이 눌렸고 변경되었고, 삭제되었는지의 정보를 가져올 수 있다는 것이었다. 따라서 Event handler 클래스를 만들고 이벤트를 처리할 수 있게 하였다.

1. <16011158 이영호>
2. <17011651 이승우>
3. <17011646 이정호>

III. 구현시의 어려움 및 해결

<17011651 이승우>

개발 환경을 똑같이 맞추는 부분부터 개발을 시작하기까지 몇번 난관이 있었지만 자주 소통을 하면서 진행하였기 때문에 비교적 짧은 시간내에 해결해나갈 수 있었다.

CalendarFx 를 활용하기 이전에는 GUI 구성 부분에서 많은 어려움이 있었다. 시간대비 퀄리티가 떨어지는 결과물이 나오는 부분도 있었고, 여러 화면을 구성하는 부분에 있어서 예상외로 삽질을 좀 많이 했던 것 같다.

프로그래밍 언어를 제외하면, 여러가지 툴의 사용은 거의 처음이나 다름없었기 때문에 전부 새로 배워가며 진행을 하느라 구현에 시간이 오래걸린 부분이 있었다.

크롤링 파트를 구현하면서 치명적인 오류가 발생하는 경우는 없었다. 하지만 코드를 작성하다보니 초기에 사용자에게 제공하고자 했던 정보들보다 더 상세한 데이터를 보기쉽게 제공하였으면 좋겠다는 욕심이 생기면서, 사용 편의성을 늘리는 것에 좀 신경을 많이 썼다. 예외처리 부분에서도 애를 많이 먹었다.

1. <16011158 이영호>
2. <17011651 이승우>
3. <17011646 이정호>

III. 구현시의 어려움 및 해결

<17011651 이승우>

또한 오류를 수정하고 세세한 기능을 다듬는 부분에 있어서는 코드에 대한 전체적인 이해가 필요했기 때문에 다른 조원들이 미리 작성한 코드를 보면서 공부하는 데에 시간이 좀 걸렸다. 모르는 부분이나 애매한 부분들은 메신저를 이용해 서로 정보를 교환해가면서 해결했다.

코로나로 인해 따로 만나서 회의를 하거나 하지는 못했지만 구글 meet이나 카카오톡 등의 메신저를 이용해서 프로젝트를 진행하였고, github를 이용하여 코드를 따로 합치거나 하는 작업을 하는 부분에 있어서 비교적 시간을 적게 들었기 때문에 github를 이용하기 이전에 해왔던 팀플들보다는 한결 수월했다고 느꼈다.

1. <16011158 이영호>
2. <17011651 이승우>
3. <17011646 이정호>

III. 구현시의 어려움 및 해결

<17011646 이정호>

이번 팀프로젝트를 하면서 많은 것들이 처음이었다.

생각해보면 java빠고는 모든 것이 처음이었다.

팀원들도 급히 조직되었고 GitHub도 처음 사용하면서 JavaFX나 Scenebuilder나 DB, 셀레니움도 사용해 본 적이 없었다.

처음에 각자 javaFX를 이용하여 GUI를 제작해보자 했던 경우에 팀원들도 어려워 했던 거 같다. GUI를 설계하는데 기능이나 작동방식들을 명확하게 하지 못하고 디자인에도 상당한 애를 먹으며 일반인 입장에서 디자인을 하려고 하니 시간이 상당히 오래 걸린다는 것을 알고 CalendarFX를 사용하기로 하였다.

제작부분을 맡은 메모기능은 Scenebuilder를 활용하여 만들었는데 이는 FXML를 활용하는 방법이다. 이 방법에 대해 한국어 메뉴얼도 굉장히 적었고 유튜브나 구글링을 통해 찾아보면서 공부를 하게 되었는데 정보가 옛날 것이고 내용의 질도 상당히 낮은 것들이 많아서 찾기가 어려웠다.

그래도 인터넷상에서 찾아 보면 해결방법을 찾을 수 있었고 코드를 작성할 수 있어서 다행이라고 생각하였다.

1. JAVA
2. SEIENIUM
3. CALENDAR FX
4. SQLITE
5. SCENE BUILDER

IV. 개발 툴 정리

1. JAVA

개발하는데 사용한 메인 언어이다.

이클립스를 사용하여 JDK를 설치하고 개발하였다.

너무나 유명하고 대중적인 언어이고, 계획서 보다 일찍 끝나서 만약 시간이 많이 남아 앱을 만들 수 있다면, 같은 JAVA로 작업하는 것이 나을 것 같아서 JAVA로 선정하게 되었다.

JAVA FX의 경우는 JAVA에서 GUI 라이브러리로 GUI 개발을 빠르게 할 수 있도록 도와준다.



1. JAVA
2. SEIENIUUM
3. CALENDAR FX
4. SQLITE
5. SCENE BUILDER



IV.개발 툴 정리

2. SEIENIUUM

블랙보드 과제와 학사일정을 알아오기 위하여 셀레니움
을 사용했다. 셀레니움은 브라우저 자동화 도구로
크롤링을 쉽게 할 수 있게 도와준다.

다만 셀레니움 사용시에 문제가 되었던 적이 있었는데
보안상의 문제로 아이디와 패스워드를 우회해서 할 시에
문제가 된다. 프로그램 구현시에 우회 로그인인 아닌 직접
로그인 방식을 구현하면 문제가 없을 것이라고 생각했고
사실상 마우스로 클릭 CTRL + C 누르는 작업과 크게 다르지
않는 작업이다.

전산정보실 <itservice@sejong.ac.kr>
"김성규", "백정우", 나에게 ▾

안녕하세요 전산정보실 입니다.

(위 메일에서는 직접 로그인 하지 않고, 다른 서버를 거쳐
로그인 시도를 하였기에, 한 서버에서 여러 유저의 로그인
시도가 있어서 문제가 되었다)

먼저 ID/PW 를 우회하여 접속하는 방법은 허용하지 않고 있습니다.

올해 하반기 중 포탈 로그인 시 2채널 로그인 보안이 활성화 될 예정이며 이런 경우 우회 로그인 자체가 불가능 합니다.

(지난학기 블랙보드 로그인 시 적용된적이 있음)

추가로 학생은 ID/PW 를 저장하지 않는다고 설명하셨습니다

애석하게도 저장여부와는 관계없이 학교 시스템 입장에선 안전하지 않은 채널로 접속하는 것이므로

이는 허용할 수 없고 접속 방법 자체가 프록시 접속 및 크롤링 방법에 해당하므로

해당 프로그램 운영은 즉시 중단하시기를 바랍니다.

감사합니다.

GIT 주소 : <https://github.com/SeleniumHQ/selenium.git>

1. JAVA
2. SEIENIUUM
3. CALENDAR FX
4. SQLITE
5. SCENE BUILDER

IV.개발 툴 정리

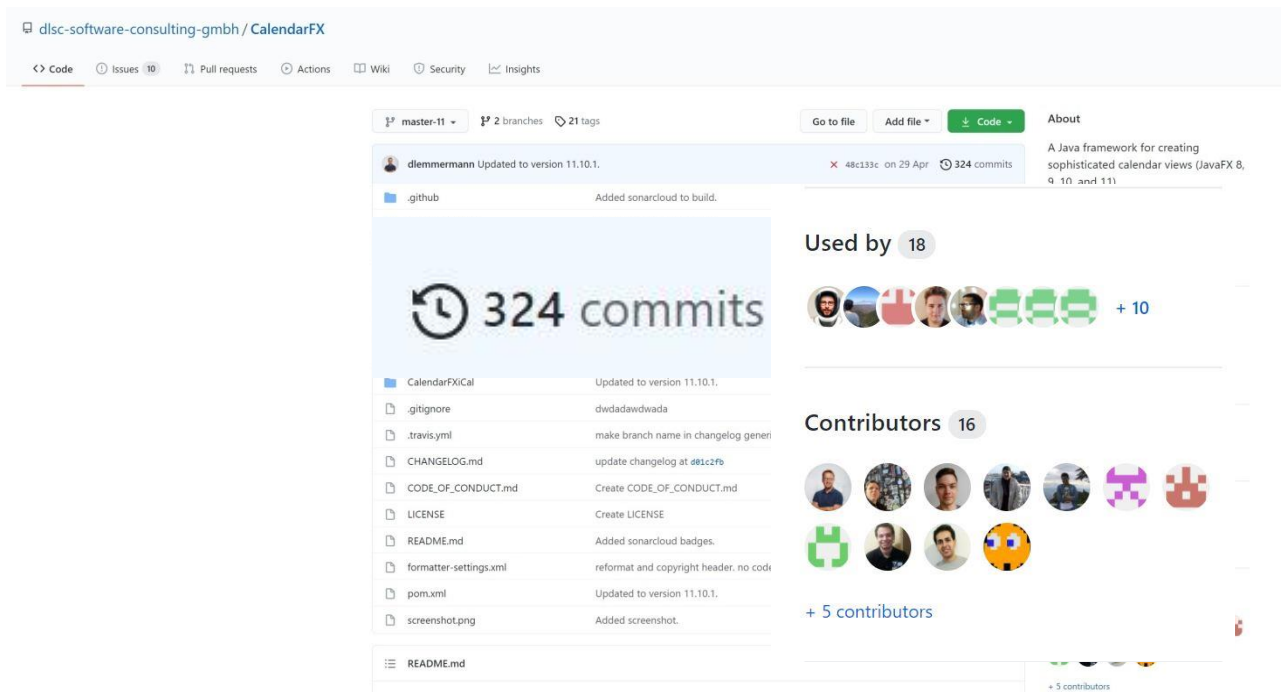
3. CALENDAR FX

CALENDAR FX 라이브러리는 원래 상업용으로 판매되던 라이브러리 였다. 하지만 수익성이 없다고 생각한 나머지 오픈소스로 깃헙에 풀게 되었다.

기존에 수익성을 목표로 만들어졌던 만큼 상당히 탄탄한 UI를 제공한다.

CALENDAR FX를 이용하면 기존에 어려웠던 UI디자인을 빠르게 해결하고, 나머지의 작업에 집중해서 작업 시간을 빠르게 단축시킬 수 있다.

라이선스는 MIT 라이선스이다.



GIT 주소 : <https://github.com/dlsc-software-consulting-gmbh/CalendarFX>

1. JAVA
2. SEIENIUM
3. CALENDAR FX
4. SQLITE
5. SCENE BUILDER



IV.개발 툴 정리

4. SQLITE

MySQL 이나 Maria db의 경우에는 배포시에 배포하는 PC에도 데이터베이스를 설치해야 사용할 수 있는데 SQLITE 는 데이터베이스를 설치하지 않아도 프로그램 내부에서의 데이터베이스 명령을 처리 할 수 있게 도와준다. 라이선스는 퍼블릭 도메인이다.

DB BROWERS 를 설치하면 SQL 쿼리문과, 데이터베이스 안에 들어있는 데이터를 확인할 수 있다.

```
try {
    stat.executeUpdate("DELETE FROM Private_Schedule;");
    stat.executeUpdate("DELETE FROM University_Schedule;");
    stat.executeUpdate("DELETE FROM BlackBord_Schedule;");
    stat.executeUpdate("DELETE FROM BlackBord_Movie_Schedule;");

    save_calendar_method(Private_Schedule_list, "Private_Schedule");
    save_calendar_method(University_Schedule_list, "University_Schedule");
    save_calendar_method(BlackBord_Schedule_list, "BlackBord_Schedule");
    save_calendar_method(BlackBord_Movie_Schedule_list, "BlackBord_Movie_Schedule");
}
```

(SQLITE 쿼리 실행 코드 일부)

테이블(T): BlackBord_Movie_Schedule

Title	Fullday	StartDate	EndDate	StartTime ▼ ¹	EndTime
필터	필터	필터	필터	필터	필터

데이터 베이스의 설정값들

Title : 제목

Full day : 하루종일 하는 일정일 경우

외, 시작 종료 시간 등의 값을 저장함

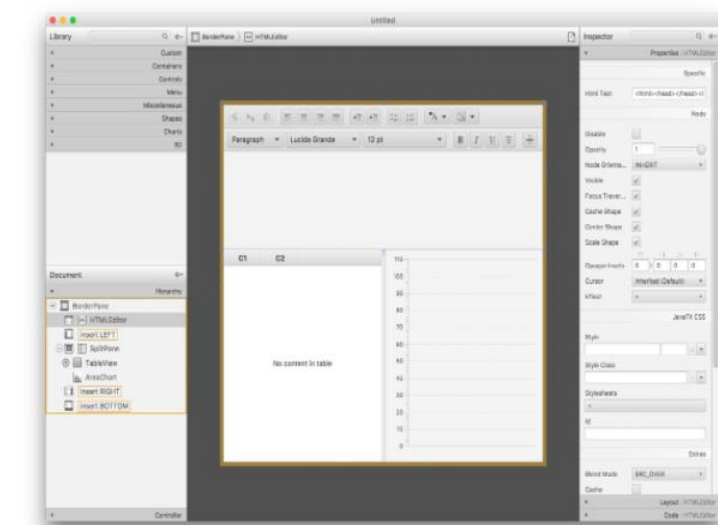
1. JAVA
2. SEIENIUM
3. CALENDAR FX
4. SQLITE
5. SCENE BUILDER

IV.개발 툴 정리

5. SCENE BUILDER

이 프로그램은 JavaFX의 GUI 디자인을 도와주는 프로그램이다. 라이선스로는 BSD 를 사용하고 있으며 코드로 적어 만들어내는 것보다 그래픽을 이용해 원하는 컨트롤러를 원하는 화면의 위치에 만들기 때문에 상당한 시간 감소효과가 생긴다. 또한 FXML를 지원하기 때문에 웹디자이너와 협업을 할 시 MVC디자인 패턴을 사용하여 이벤트 핸들과 디자인부분의 분리가 확실해 작업효율이 증가하는 효과를 보여준다.

Scene Builder



컨트롤러

작업 화면

컨트롤러 변수

1. GITHUB 주소 및 기여도

V. 결론

1. GITHUB 주소 및 기여도

■ Github 주소

<https://github.com/YOUNGHOO/OPENSOURCE16>

기여도

<16011158 이영호>

Calendarfx ui 를 이용하여 UI 기능을 추가하였고
UI 이벤트 처리(삭제, 변경, 등)를 하는 Event_handle.java
데이터 베이스 처리를 하는 Data_base.java, 와 Main.java의
일부분과, 좌측 버튼 이벤트를 감지하는App_Controller.java
를 작성했습니다.

<17011646 이정호>

SceneBuilder를 이용하여 메모기능 담당했습니다.
MemoPaneController.java 소스코드를 작성했습니다.

<17011651 이승우>

Selenium을 이용한 블랙보드, 학사일정 데이터
크롤링 및 파싱 파트를 담당했습니다.
SeleniumCrawl.java 소스 코드를 작성하였습니다.
해당 파트 구현 이후, 일정 동기화 스레드 처리,
초기화 버튼 구현 및 오류 수정 등의 작업을 하였습니다.

기여도 순위 (10을 기반으로)

1. <16011158 이영호> 4
2. <17011651 이승우> 3.5
3. <17011646 이정호> 2.5