


200711429 박상욱

200711439 송인하

200711444 양동은 

Team Presentaion #1

Plan & Analysis



Contents

1 - 1

Define Draft Plan

2 - 1

Define Conceptual Model

1 - 2

**Create Preliminary
Investigation Report**

2 - 2

Define Sequence Diagram

1 - 3

Define Requirements

2 - 3

Define Operation Contract

1 - 4

Define Business Concept and Model



1-1. Define Draft Plan

Step1. Motivation and Objective of project

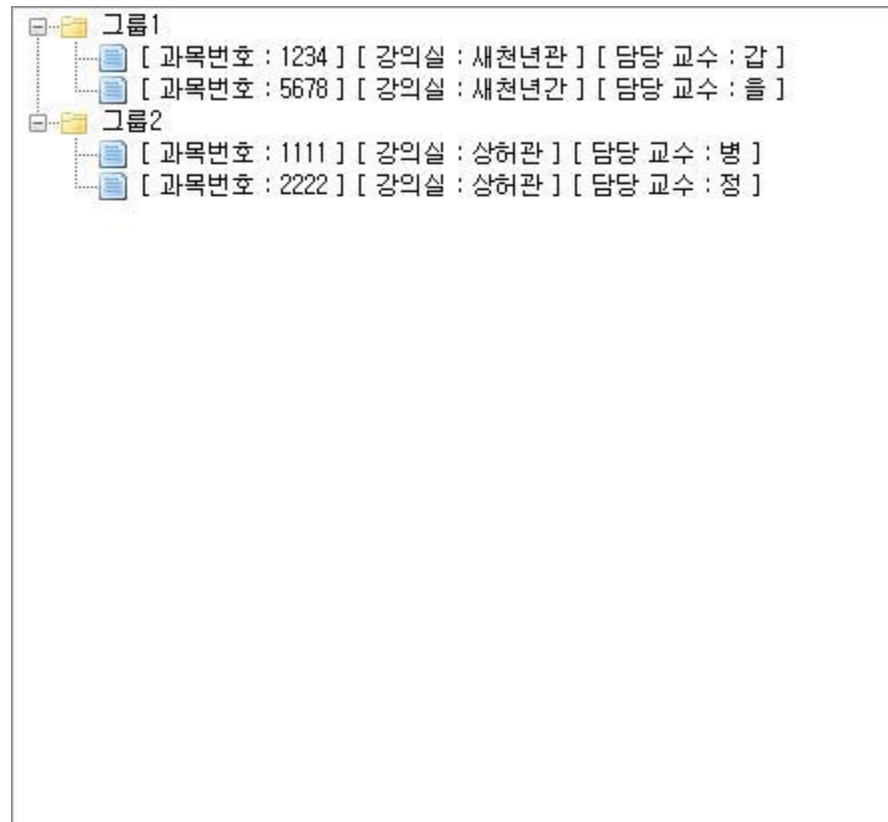
1-1. Motivation

매 학기 수강신청을 하기 위해서는 자신이 꼭 들어야 하는 필수과목과 개인적으로 듣고 싶은 희망 과목으로 나누어 경우의 수를 계산하여 제작하여야 한다. 모든 경우의 수를 사용자가 직접 계산하기 위해서는 많은 시간과 노력이 필요하며, 모든 경우의 수를 계산했다는 보장을 받지 못한다.

이러한 학생들의 고충을 해결하기 위해 경우의 수를 계산해주는 시스템을 구축한다. 또한 이러한 상황 발생은 학기마다 발생하므로 시스템 사용 빈도가 낮기 때문에 따로 시스템을 배우지 않고 쉽고 빠르게 만들 수 있는 시스템으로 구축한다.

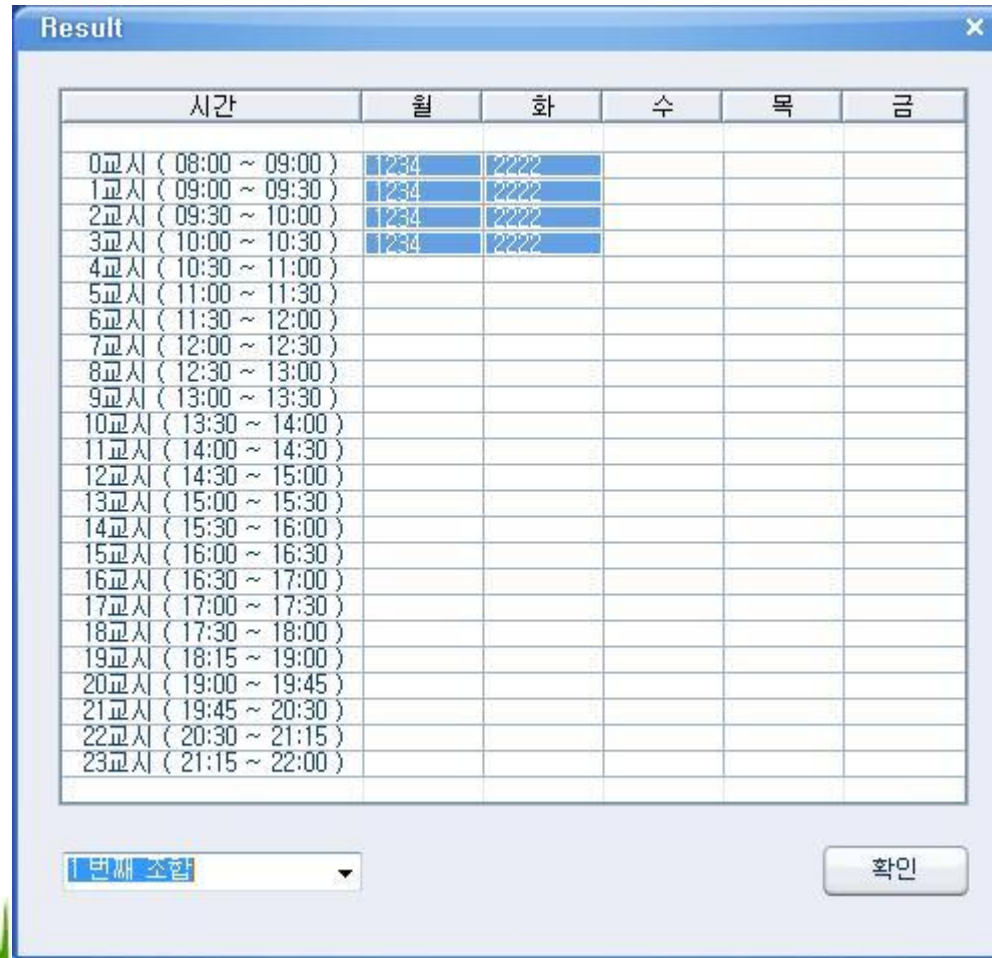
0. Example

Step1. Make Groups end Subjects



0. Example

Step2. Check Result

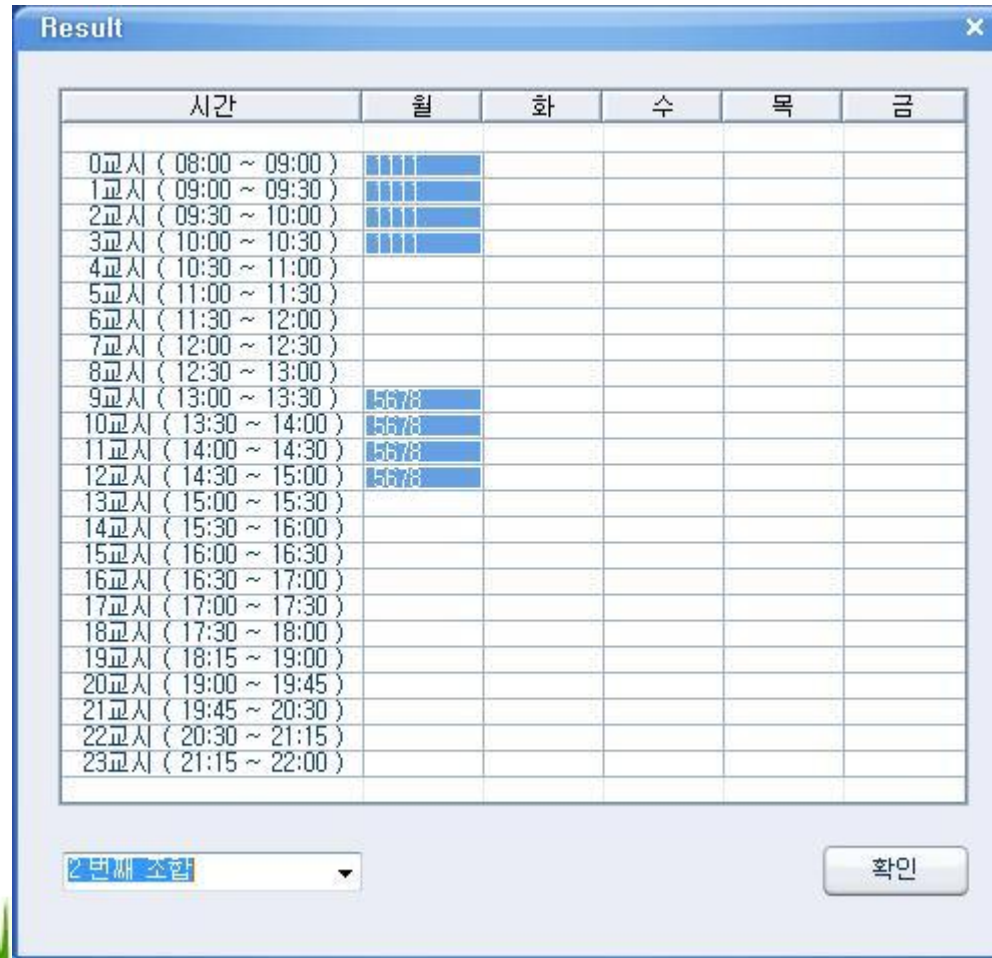


The screenshot shows a window titled "Result" with a close button (X) in the top right corner. The window contains a table with 6 columns: "시간" (Time), "월" (Month), "화" (Tuesday), "수" (Wednesday), "목" (Thursday), and "금" (Friday). The rows list 24 class periods from 0교시 to 23교시. The first four rows (0교시 to 3교시) have blue highlights in the "월" and "화" columns, with the value "2222" in the "화" column. The remaining rows are empty. At the bottom left, there is a dropdown menu with the text "1 번째 조합" and a downward arrow. At the bottom right, there is a button labeled "확인" (Confirm).

시간	월	화	수	목	금
0교시 (08:00 ~ 09:00)	2222	2222			
1교시 (09:00 ~ 09:30)	2222	2222			
2교시 (09:30 ~ 10:00)	2222	2222			
3교시 (10:00 ~ 10:30)	2222	2222			
4교시 (10:30 ~ 11:00)					
5교시 (11:00 ~ 11:30)					
6교시 (11:30 ~ 12:00)					
7교시 (12:00 ~ 12:30)					
8교시 (12:30 ~ 13:00)					
9교시 (13:00 ~ 13:30)					
10교시 (13:30 ~ 14:00)					
11교시 (14:00 ~ 14:30)					
12교시 (14:30 ~ 15:00)					
13교시 (15:00 ~ 15:30)					
14교시 (15:30 ~ 16:00)					
15교시 (16:00 ~ 16:30)					
16교시 (16:30 ~ 17:00)					
17교시 (17:00 ~ 17:30)					
18교시 (17:30 ~ 18:00)					
19교시 (18:15 ~ 19:00)					
20교시 (19:00 ~ 19:45)					
21교시 (19:45 ~ 20:30)					
22교시 (20:30 ~ 21:15)					
23교시 (21:15 ~ 22:00)					

0. Example

Step2. Check Result



The screenshot shows a window titled "Result" with a table of class times and counts. The table has six columns: "시간" (Time), "월" (Month), "화" (Tuesday), "수" (Wednesday), "목" (Thursday), and "금" (Friday). The rows list class times from 0교시 (08:00 ~ 09:00) to 23교시 (21:15 ~ 22:00). The "월" column contains blue bars representing counts for the first 12 classes, and the "수" column contains the number "5678" for classes 9 through 12. At the bottom, there is a dropdown menu with "2 번째 조합" and a "확인" button.

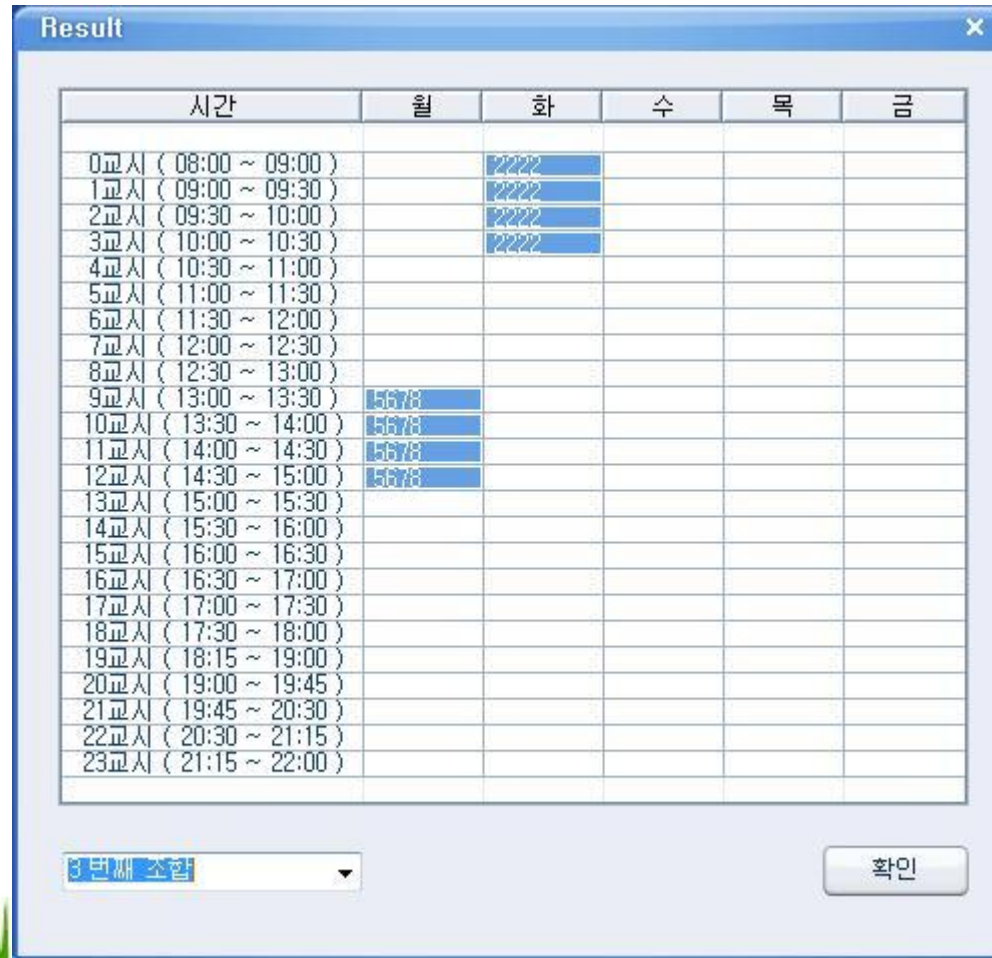
시간	월	화	수	목	금
0교시 (08:00 ~ 09:00)	■■■■■				
1교시 (09:00 ~ 09:30)	■■■■■				
2교시 (09:30 ~ 10:00)	■■■■■				
3교시 (10:00 ~ 10:30)	■■■■■				
4교시 (10:30 ~ 11:00)					
5교시 (11:00 ~ 11:30)					
6교시 (11:30 ~ 12:00)					
7교시 (12:00 ~ 12:30)					
8교시 (12:30 ~ 13:00)					
9교시 (13:00 ~ 13:30)	5678				
10교시 (13:30 ~ 14:00)	5678				
11교시 (14:00 ~ 14:30)	5678				
12교시 (14:30 ~ 15:00)	5678				
13교시 (15:00 ~ 15:30)					
14교시 (15:30 ~ 16:00)					
15교시 (16:00 ~ 16:30)					
16교시 (16:30 ~ 17:00)					
17교시 (17:00 ~ 17:30)					
18교시 (17:30 ~ 18:00)					
19교시 (18:15 ~ 19:00)					
20교시 (19:00 ~ 19:45)					
21교시 (19:45 ~ 20:30)					
22교시 (20:30 ~ 21:15)					
23교시 (21:15 ~ 22:00)					

2 번째 조합

확인

0. Example

Step2. Check Result




The screenshot shows a window titled "Result" with a close button (X) in the top right corner. The window contains a table with 6 columns: "시간" (Time), "월" (Month), "화" (Tuesday), "수" (Wednesday), "목" (Thursday), and "금" (Friday). The rows list 24 class periods from 0교시 to 23교시. The "화" column contains "2222" for periods 0-4 and "5678" for periods 9-12. A dropdown menu at the bottom left shows "3 번째 조합" (3rd combination) and a "확인" (Confirm) button is at the bottom right.

시간	월	화	수	목	금
0교시 (08:00 ~ 09:00)		2222			
1교시 (09:00 ~ 09:30)		2222			
2교시 (09:30 ~ 10:00)		2222			
3교시 (10:00 ~ 10:30)		2222			
4교시 (10:30 ~ 11:00)					
5교시 (11:00 ~ 11:30)					
6교시 (11:30 ~ 12:00)					
7교시 (12:00 ~ 12:30)					
8교시 (12:30 ~ 13:00)					
9교시 (13:00 ~ 13:30)	5678				
10교시 (13:30 ~ 14:00)	5678				
11교시 (14:00 ~ 14:30)	5678				
12교시 (14:30 ~ 15:00)	5678				
13교시 (15:00 ~ 15:30)					
14교시 (15:30 ~ 16:00)					
15교시 (16:00 ~ 16:30)					
16교시 (16:30 ~ 17:00)					
17교시 (17:00 ~ 17:30)					
18교시 (17:30 ~ 18:00)					
19교시 (18:15 ~ 19:00)					
20교시 (19:00 ~ 19:45)					
21교시 (19:45 ~ 20:30)					
22교시 (20:30 ~ 21:15)					
23교시 (21:15 ~ 22:00)					

1-1. Define Draft Plan


Step1. Motivation and Objective of project

1-2. Object of Project

- 듣고 싶은 과목을 입력하면 가능한 **모든 경우의 수**를 시간표 형식으로 볼 수 있다.
 - 경우의 수를 선택하면 수강신청 시 **인원제한 수가 적은 순서대로** 신청할 수 있게 리스트를 볼 수 있다.
 - 선택한 시간표를 기반으로 자신의 학기 시간표를 **그림파일**로 볼 수 있다.
 - 모든 시스템은 **직관적**이며 따로 배우지 않고도 **쉽게** 사용할 수 있다.
- 

1-1. Define Draft Plan

Step2. Functional Requirements


- 시간표 DB에서 원하는 과목 선택
 - 자신이 선택한 과목 저장
 - 저장한 과목 불러오기
 - 선택한 과목들로 가능한 시간표 모든 경우의 수 출력
 - 선택한 경우의 수로 수강신청 시 참고 할 수 있는 리스트 txt 생성
 - 신청 완료 후 그림으로 시간표 저장 가능
- 

1-1. Define Draft Plan

Step3. Non-functional Requirements

- 사용자들이 쉽게 사용할 수 있도록 직관적으로 설계한다.
- 모든 경우의 수 계산은 10초 이내로 한다.

Step4. Estimate resources


- **Human Resource : 3명**
설계 및 분석, 구현 모든 단계를 3명이 같이 진행한다.
 - **Project Duration : 4주**
- 

1-2. Create preliminary Investigation Report

Step1. Alternative Solutions

- 쉽게 사용할 수 있는 직관적인 GUI 개발을 위해 MFC를 사용하여 구현 가능
- 과거에 제작했었던 시간표 관련 시스템 구축을 경험으로 더욱 유연한 시스템 구축

Step2. Project's Justification

- OOAD 적용을 통한 프로젝트 개발 경험이 없는 현재 개발자로서 분석부터 설계, 구현까지 객체지향 개발 방법론을 통한 구현경험은 추후 실무에 크게 작용할 수 있다.
 - MFC를 통하여 GUI를 개발함에 있어서 시스템 구축에 크게 작용하고 있는 GUI 요소에 대하여 공부 할 수 있다.
- 

1-2. Create preliminary Investigation Report

Step3. Risk Management

Risk	Probability	Significance	Weight
OOAD 적용의 첫 사례	4	5	20
UML 적용의 첫 사례	3	5	15
개발 시간 부족	3	3	9
팀 간의 의사소통	1	2	2
MFC 관련 지식 부족	2	2	4

1-2. Create preliminary Investigation Report

Step4. Risk Reduction Plan

- **OOAD 적용의 첫 사례 (20)**
 - 과제 및 도서관 서적, 수업 자료로 충분한 공부를 한다.
- **UML 적용의 첫 사례 (15)**
 - 과제 및 도서관 서적, 수업 자료로 충분한 공부를 한다.
- **개발 시간 부족 (9)**
 - 수업시간 외의 모든 시간을 쏟아 진행한다.
- **팀 간의 의사소통 (2)**
 - 서로의 의견을 들으며 충분히 토의 후 결정한다.
- **MFC 관련 지식 부족 (4)**
 - GUI 관련 부분만 빠르게 공부한다.

1-3. Define Requirements

1. 소개(Introduction)

가. 목적 (Purpose)

본 문서는 Window 기반의 Time Table 개발을 위한 소프트웨어 요구사항 명세서이다. 학생들이 수강신청을 하기 위해 원하는 과목 및 시간대를 선택하는데 보다 쉽고 빠르고 직관적으로 작성하기 위한 시스템을 설계 및 구현한다. 시스템 구현에 있어서 사용자 및 이해관계자, 시스템 요구사항을 명확히 도출해 개발에 있어 체계적인 진행을 함으로써 시간적인 비용을 최소화하고, 정확한 요구사항 분석으로 고객의 만족을 최대화하는 것이 이 문서의 목표이다.

나. 제품 범위 (Product Scope)

기존 수강신청 전 시간표를 작성하기 위해서는 학생들은 직접 손으로 필수 과목과 선택과목을 선택하여 경우의 수를 계산해가며 작성해야만 했다. 이러한 불편함을 해소하고자, 시스템이 원하는 과목을 선택하면 가능한 모든 경우의 수를 출력해줌으로써 시간표 제작에 있어서 빠르고 간편하게 작성 할 수 있도록 한다.

1-3. Define Requirements

1. 소개(Introduction)

다. 용어 및 약어 정의 (Definitions, Acronyms, and Abbreviations)

Term	Definition
DB	Data Base
GUI	Graphic User Interface
UML	Unified Modeling Language

1-3. Define Requirements

2. 일반적인 기술 (General Description)

가. 제품 주요 기술 (Product Function)

사용자가 원하는 과목을 선택한 후 결과 수행 버튼을 누르면 시간표의 가능한 경우의 수를 모두 표현해준다.

나. 사용자 특성 (User Characteristics)

본 시스템의 사용자는 대학교에 다니는 학생을 대상으로 가정한다. 기본적으로 컴퓨터를 사용함에 있어서 무리가 없을 정도의 교육을 받은 사용자로 가정한다.



1-3. Define Requirements

2. 일반적인 기술 (General Description)

다. 제약 조건 (Constraints)

본 장의 가절에서 언급한 내용을 바탕으로 시스템을 설계하고 구현한다. 시스템의 디자인은 개발자가 선호하는 방향으로 선택 설계하되 기본 인터페이스는 논의한 결과를 바탕으로 구현한다. 다음의 항목을 준수한다.

- 사용되는 기술은 이미 널리 성능이 입증된 것을 사용한다.
- 별도의 라이선스가 필요하거나 저작권료를 지불이 필요한 기술 또는 소프트웨어의 사용을 피한다.
- 사용자에게 보다 친숙하고 편리한 방향으로 결정한다.
- 기본의 다른 시스템들과는 다른 특성화된 시스템을 구축한다.

라. 가정과 의존성 (Assumptions and Dependencies)

본 문서의 시스템은 윈도우 기반으로 설계 및 구현할 것을 가정하여 작성되었다. 따라서 모든 내용은 윈도우를 기준으로 작성되었으며, 타 운영체제에서 동일하게 적용되지 않을 수 있다.

1-3. Define Requirements

3. 가정과 의존성 (Assumptions and Dependencies)

가. 기능적 요구사항 (Functional Requirements)

ID	요구 사항 기능 내역	설명	중요도
A-1	DB 불러오기	사용자가 선택한 리스트는 불러올 수 있어야 한다.	A
A-2	DB 내보내기	사용자가 선택한 리스트는 저장할 수 있어야 한다.	A
A-3	그룹 추가 / 삭제	태그 할 수 있는 과목들은 그룹으로 관리한다.	S
A-4	과목 추가 / 삭제	그룹들에 과목을 추가 / 삭제 할 수 있다.	S
A-5	경우의 수 보기	시간표 모든 경우의 수를 출력한다.	S
A-6	리스트 저장	선택한 과목들을 txt 파일로 내보낸다.	B
A-7	그림파일로 저장	선택한 과목들의 시간표를 그림파일로 내보낸다.	B

1-3. Define Requirements

3. 가정과 의존성 (Assumptions and Dependencies)

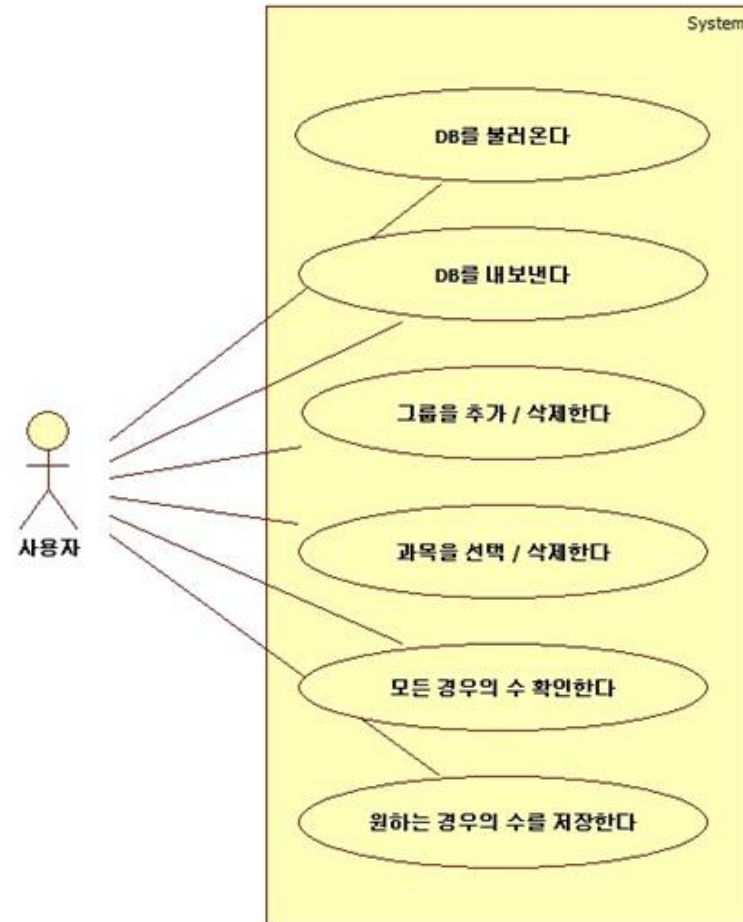
나. 비 기능적 요구사항 (Non-functional Requirements)

ID	요구 사항 기능 내역	설명	중요도
R-8	사용 편리성	처음 사용하는 사용자도 사용이 쉬워야 한다.	S
R-9	성능	경우의 수는 10초 이내에 모두 완료되어야 한다.	A
R-10	신뢰성	가능한 모든 경우의 수를 출력해주어야 한다.	S
R-11	중복 처리	같은 그룹이나 다른 그룹에서 과목은 중복 선택될 수도 있다.	A

1-3. Define Requirements

4. Use Case Model

가. System Use Cases



1-3. Define Requirements

4. Use Case Model

나. Use Cases Summary

Use case ID	Use case name	설명
UC-1	DB 불러오기	사용자가 선택한 과목들 DB를 불러온다.
UC-2	DB 내보내기	사용자가 선택한 과목들 DB를 내보낸다.
UC-3	그룹 추가/삭제	특정 과목들을 묶을 수 있는 그룹을 관리한다.
UC-4	과목 선택/삭제	그룹에 추가될 과목을 관리한다.
UC-5	경우의 수 확인	해당 과목의 모든 경우의 수를 확인한다.
UC-6	선택 경우 저장	선택한 과목의 경우의 수를 txt 파일과 그림으로 저장한다.

1-3. Define Requirements

4. Use Case Model

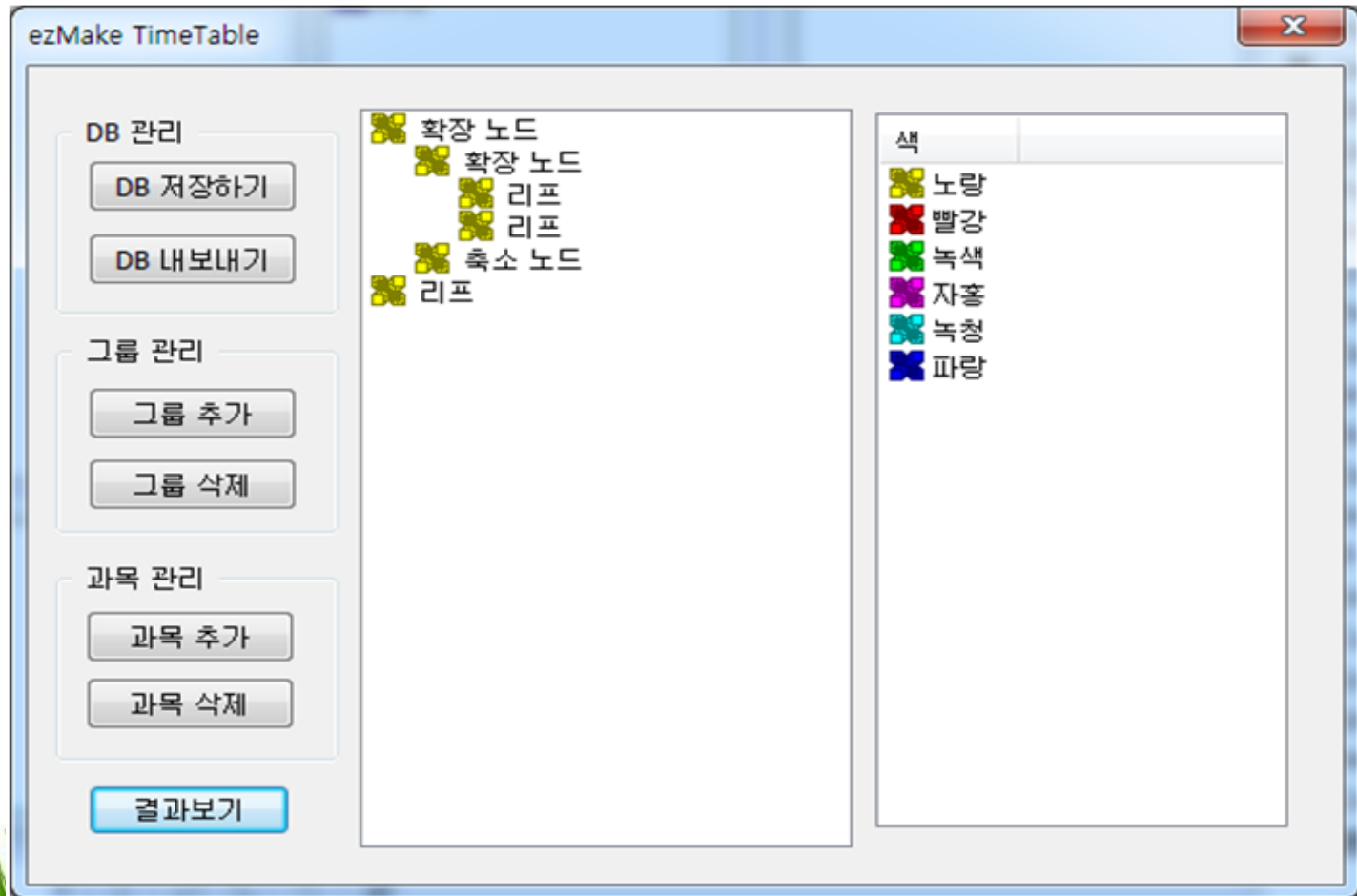
다. Detail Use Cases Scenario

Use case Description	
Use case ID	UC-5
Use case Name	경우의 수 확인
Actor	사용자
설명	해당 과목의 모든 경우의 수를 확인한다
사전조건	사용자가 원하는 그룹과 과목이 모두 리스트에 추가되어있다
사후조건	선택한 과목에 대한 모든 경우의 수가 출력된다
흐름도	<p>1.0 사용자가 리스트에 원하는 그룹과 과목을 추가한다</p> <ol style="list-style-type: none">1. 결과보기 버튼을 누른다2. 결과 확인 Form을 보여준다3. Form에 있는 리스트박스에 모든 경우의 수가 출력된다4. 리스트박스에 있는 경우의 수를 누르면 화면에 시간표가 출력된다

1-3. Define Requirements

5. 외부 인터페이스 요구사항(External Interface Requirements)

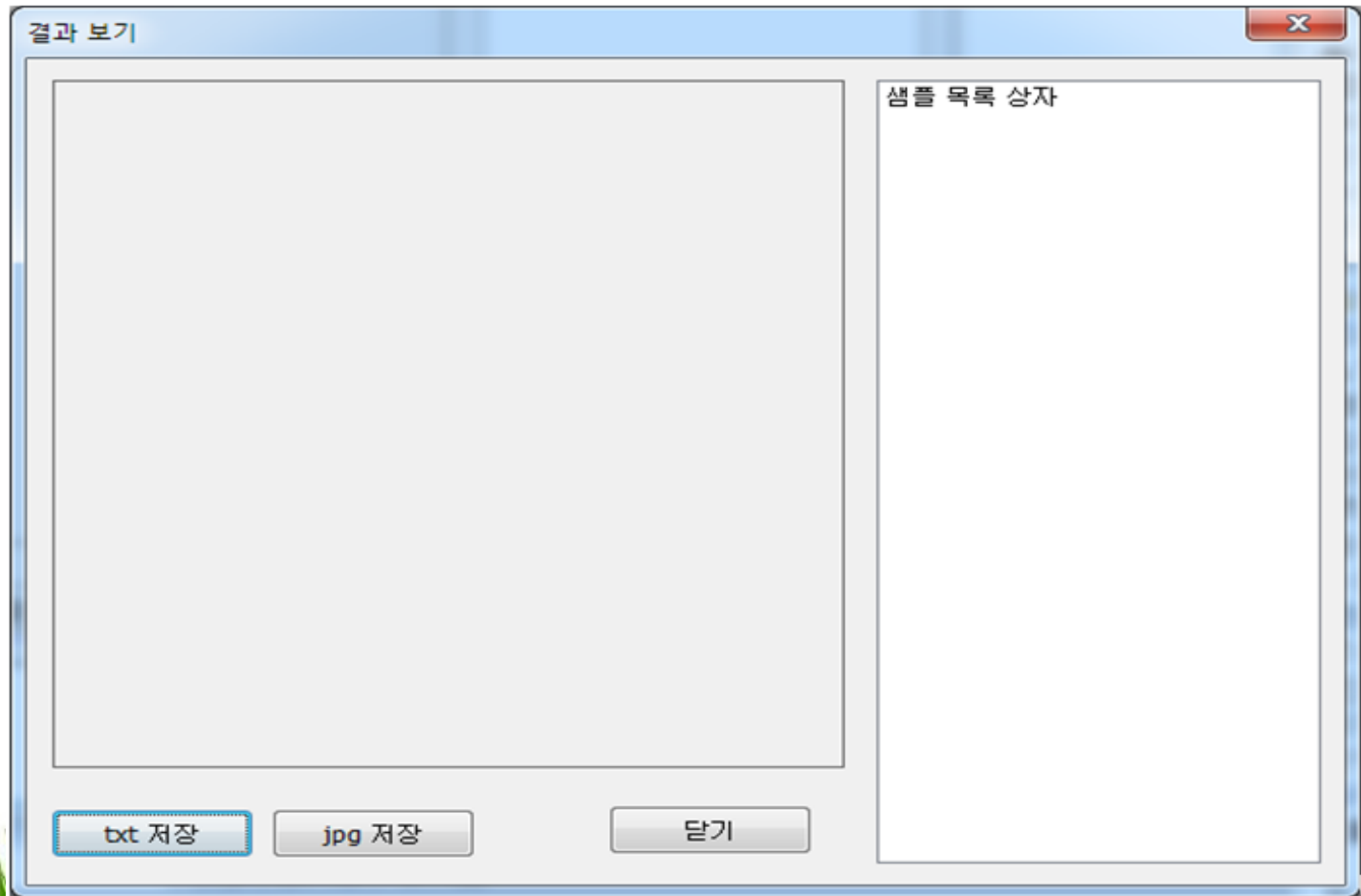
가. User Interface – 실행 GUI



1-3. Define Requirements

5. 외부 인터페이스 요구사항(External Interface Requirements)

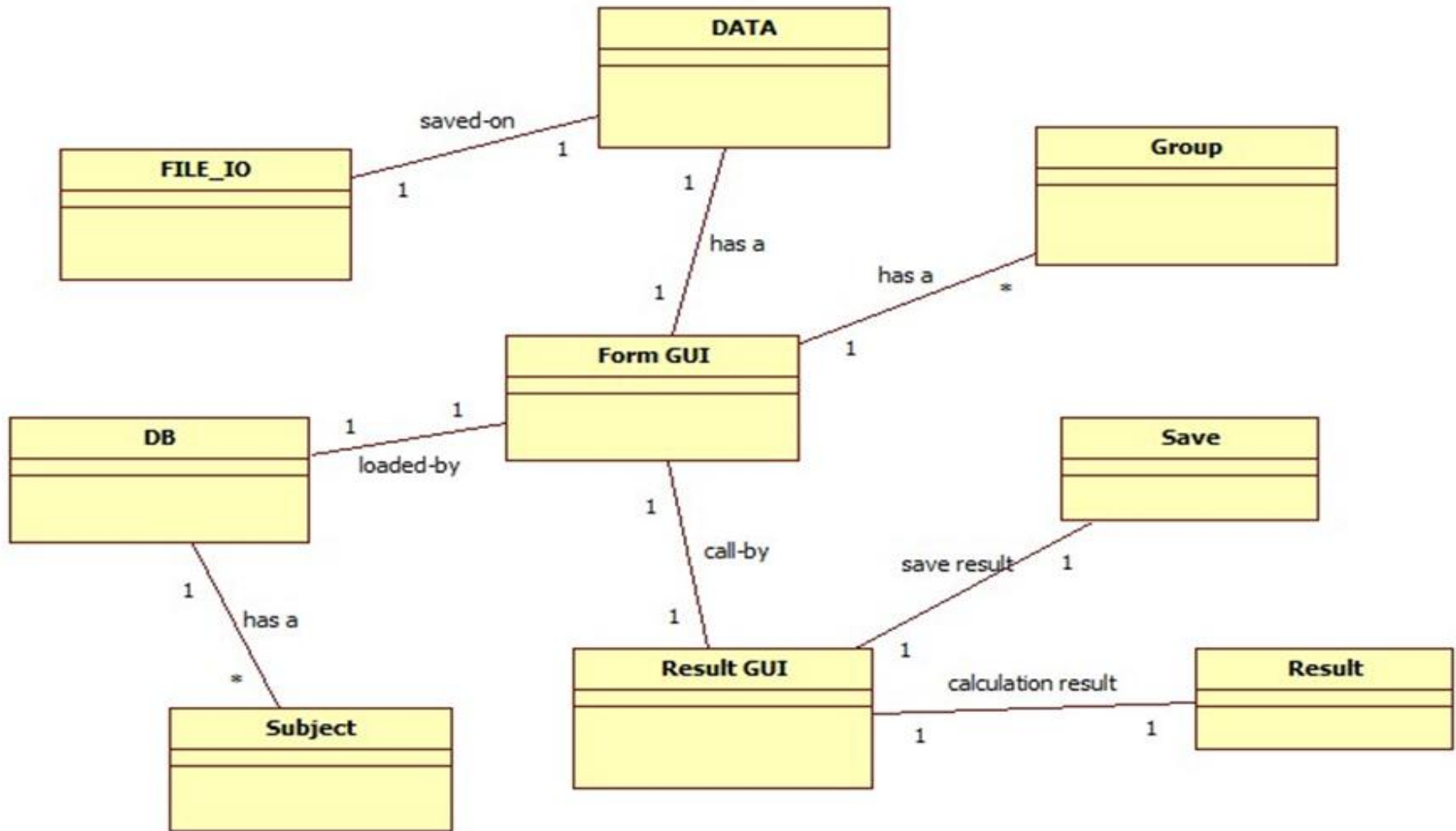
가. User Interface - 결과보기 GUI



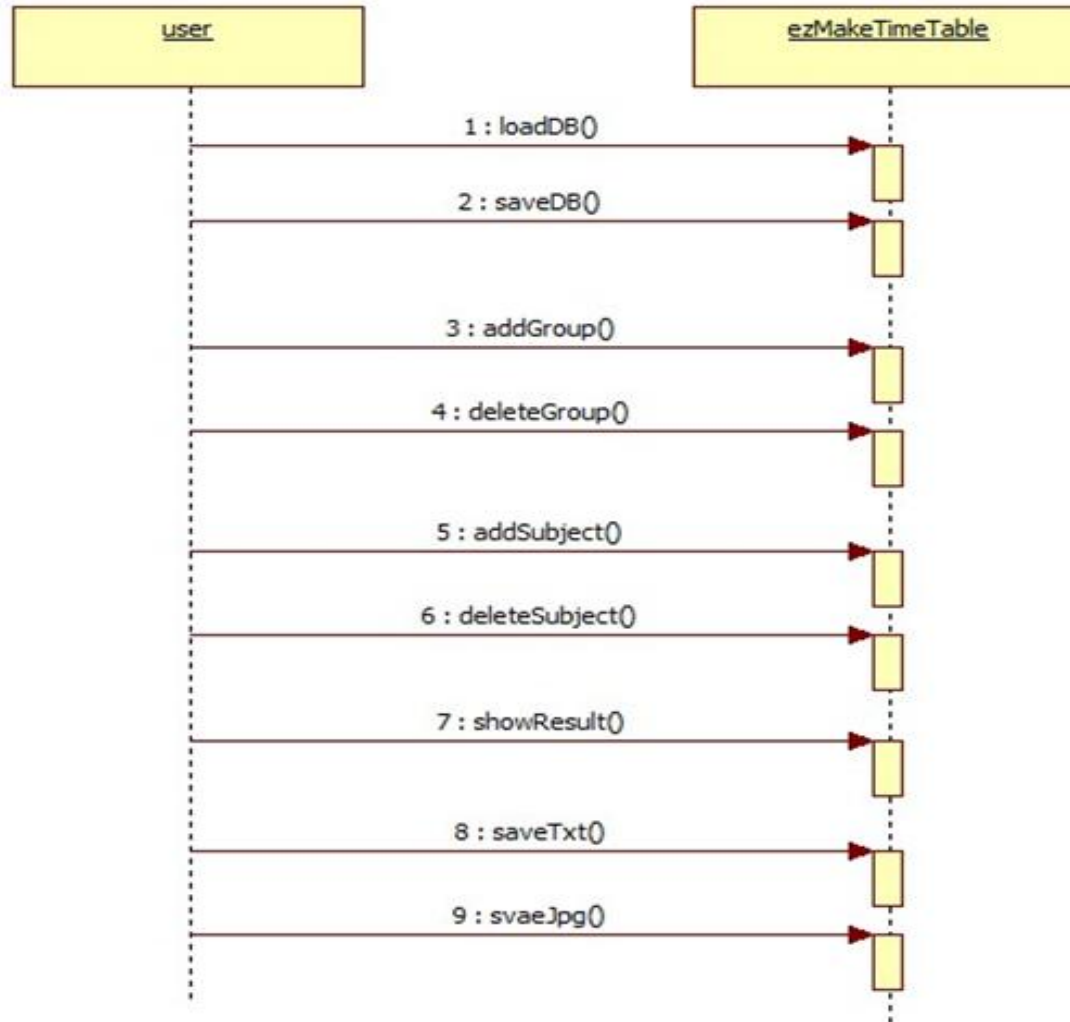
1-4. Define Business Concept and Model



2-1. Define Conceptual Model

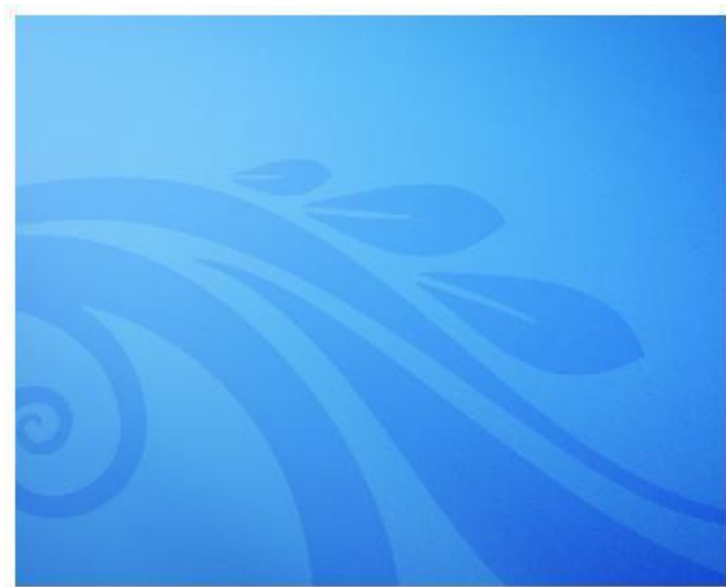


2-2. Define Sequence Diagram



2-3. Define Operation Contract

항목 ↙	의미 ↙
이름 ↙	<code>showResult(data : Data)</code> ↙
책임 ↙	<u>ResultForm</u> 을 보여주며 경우의 수 계산 결과를 <u>확인</u> 할 수 있다 ↙
유형 ↙	시스템 ↙
참조 ↙	시스템 기능 : A-5 ↙ <u>유스 케이스</u> : UC-5 ↙
주석 ↙	Data를 넘겨 주기 전에 Data는 <u>최신화</u> 를 해야 한다 ↙
예외조건 ↙	해당사항 없음 ↙
출력 ↙	<u>ResultForm</u> 이 보여주며 경우의 수를 계산하여 리스트에 보여준다 ↙
사전조건 ↙	해당사항 없음 ↙
사후조건 ↙	<ul style="list-style-type: none"> <input type="checkbox"/> Form에 인스턴트로 존재하는 Group을 전달하여 Data의 <u>오퍼레이션</u>을 호출한다 ↙ <input type="checkbox"/> Data는 전달받은 Group으로 자신의 Data를 <u>최신화</u>한다 ↙ <input type="checkbox"/> Form의 인스턴트로 존재하는 <u>ResultForm</u>을 Data를 전달하며 생성 호출한다 ↙ <input type="checkbox"/> Data를 전달받은 <u>ResultForm</u>은 Result에 Data를 전달하여 <u>계산</u>을 요청한다 ↙ <input type="checkbox"/> Result는 결과를 계산하여 <u>ResultForm</u>의 리스트의 결과를 보여준다 ↙



Thank You

