

Convenience Timetable Design

Team 4



Contents

1. Introduction

2. Decomposition description

3. Dependency description

4. Interface description

5. Detailed design description

1. Introduction

- Purpose
 - Use case 및 SR로 얻은 Requirement를 적용해 실제 구현에 앞서 시스템의 구조를 설계
 - Software 재사용 및 확장을 위해 OOP기법 및 Design pattern을 이용

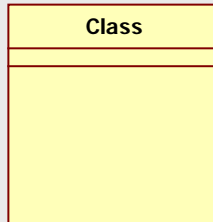
2. Decomposition description

Use Case

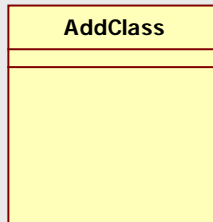
빨간색 : class 후보

1. 사용자가 프로그램을 실행 한다
2. 사용자는 저장된 시간표를 불러오거나 새로이 작성(초기화)한다.
 - 2-1 사용자가 시간표를 새로이 작성한다.
 - 2-1-1 사용자는 등록하고 싶은 시간의 칸을 선택한다.(right click)
 - 2-1-1-1 사용자가 선택한 칸에 비어있다.
 - 2-1-1-1-1 '강의 등록' 을 선택한다.
 - 2-1-1-1-1-1 강의명을 메뉴에서 선택한다.(팝업창)
 - 2-1-1-1-1-2 입력을 다했으면 '확인' 버튼을 누른다.
 - 2-1-1-1-2-3 시간표에 강의가 추가된다.
 - 2-1-1-1-2 '일정 등록' 을 선택한다.
 - 2-1-1-1-2-1 활동명, 장소, 시간을 입력한다.(팝업창)
 - 2-1-1-1-2-2 입력을 다했으면 '확인' 버튼을 누른다.
 - 2-1-1-1-2-3 시간표에 일정이 추가된다.
 - 2-1-2 사용자가 선택한 칸에 일정이 있다.
 - 2-1-2-1 '변경' 을 선택한다.
 - 2-1-2-2 '삭제' 를 선택한다.
 - 2-1-2-3 '강의 정보' 를 선택한다.
 - 2-1-2-3-1 선택한 강의에 대한 강의 정보 창이 뜬다. (인터넷 팝업창)
 - 2-2 사용자가 저장된 시간표를 불러온다.
 - 2-2-1 위의 내용과 동일한 작업 반복
3. 사용자가 시간표를 저장한다.
4. 사용자가 '수강 신청' 버튼을 클릭한다.
 - 4-1 건국대학교 수강신청 팝업창(인터넷 팝업창)이 뜬다.
5. 사용자가 '전체 강의 검색' 버튼을 클릭한다.
 - 5-1 건국대학교 종합강의시간표(인터넷 팝업창)가 뜬다.
6. 사용자가 시간표를 종료한다.

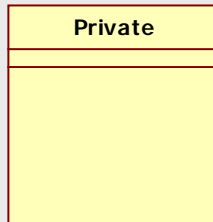
2. Decomposition description



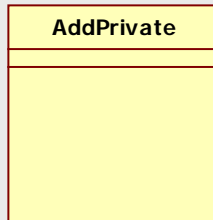
Class :: 강의명, 강의실, 교수명,
시간 정보를 가지는 클래스



AddClass :: 강의 등록 팝업창

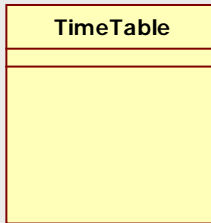


Private :: 개인 일정

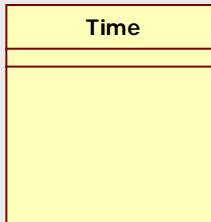


AddPrivateate :: 개인 일정 등록 팝업창

2. Decomposition description



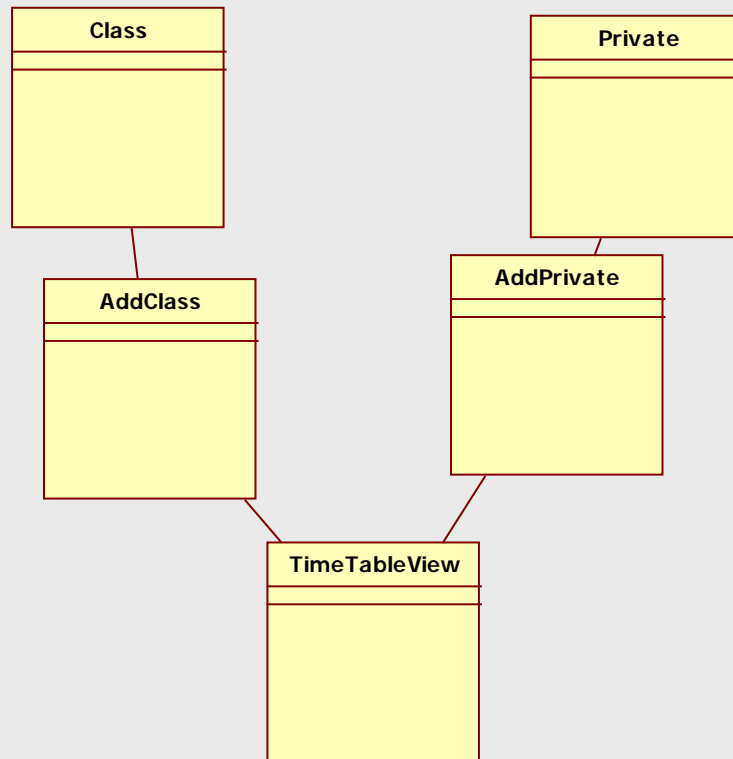
TimeTable :: 시간표



Time :: (시간) 칸

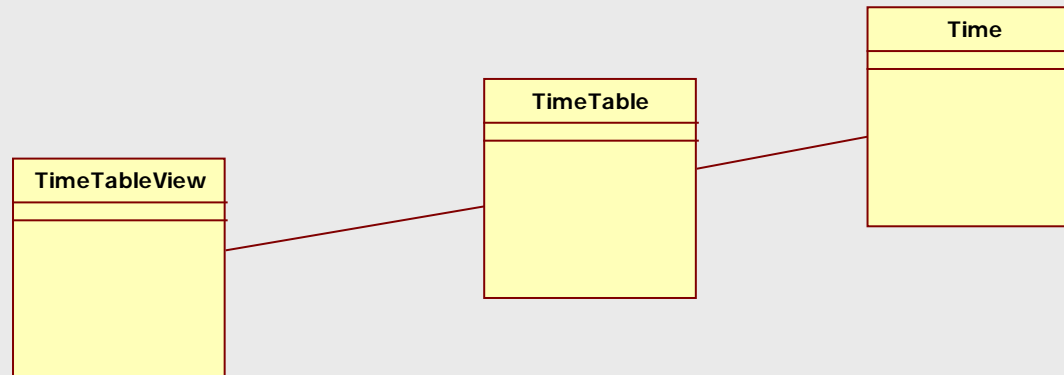
3. Dependency description

- AddPrivate, AddClass, TimeTableView



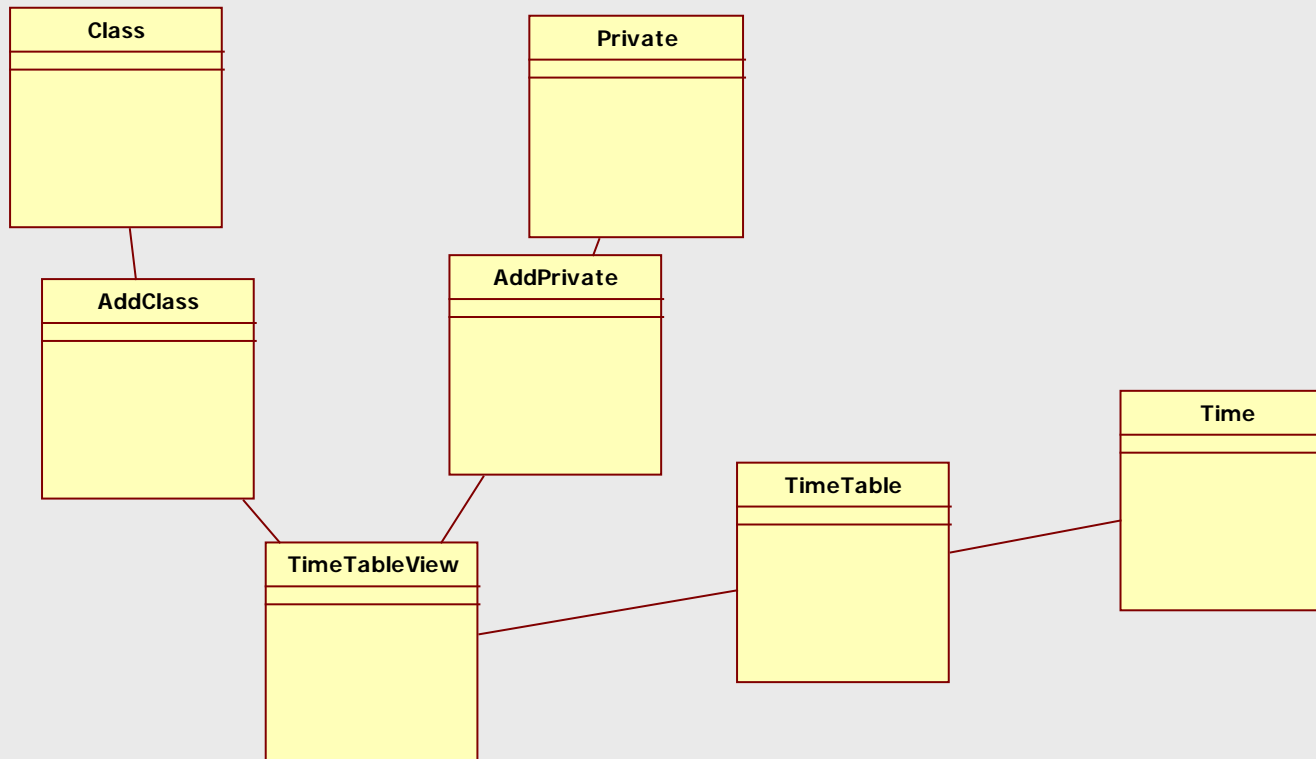
3. Dependency description

- AddPrivate, AddClass, TimeTableView



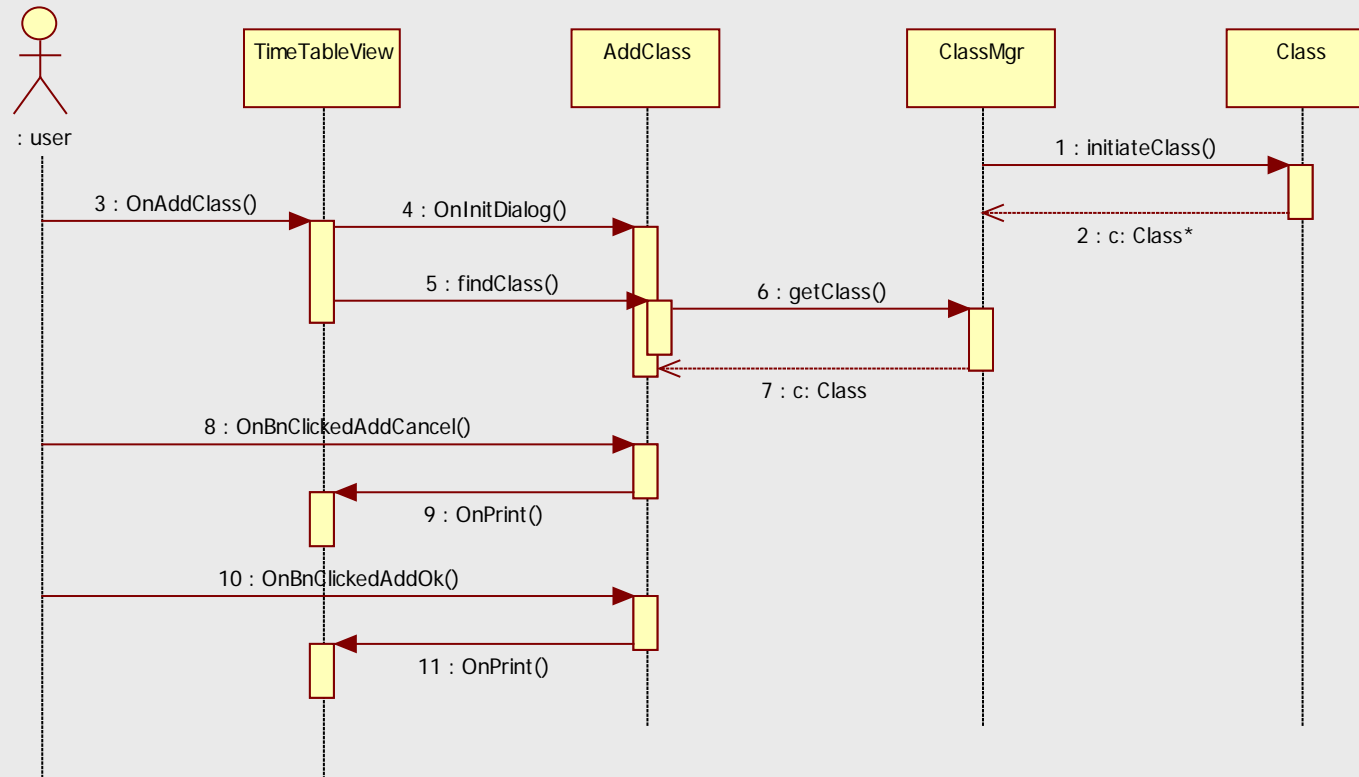
3. Dependency description

- AddPrivate, AddClass, TimeTableView



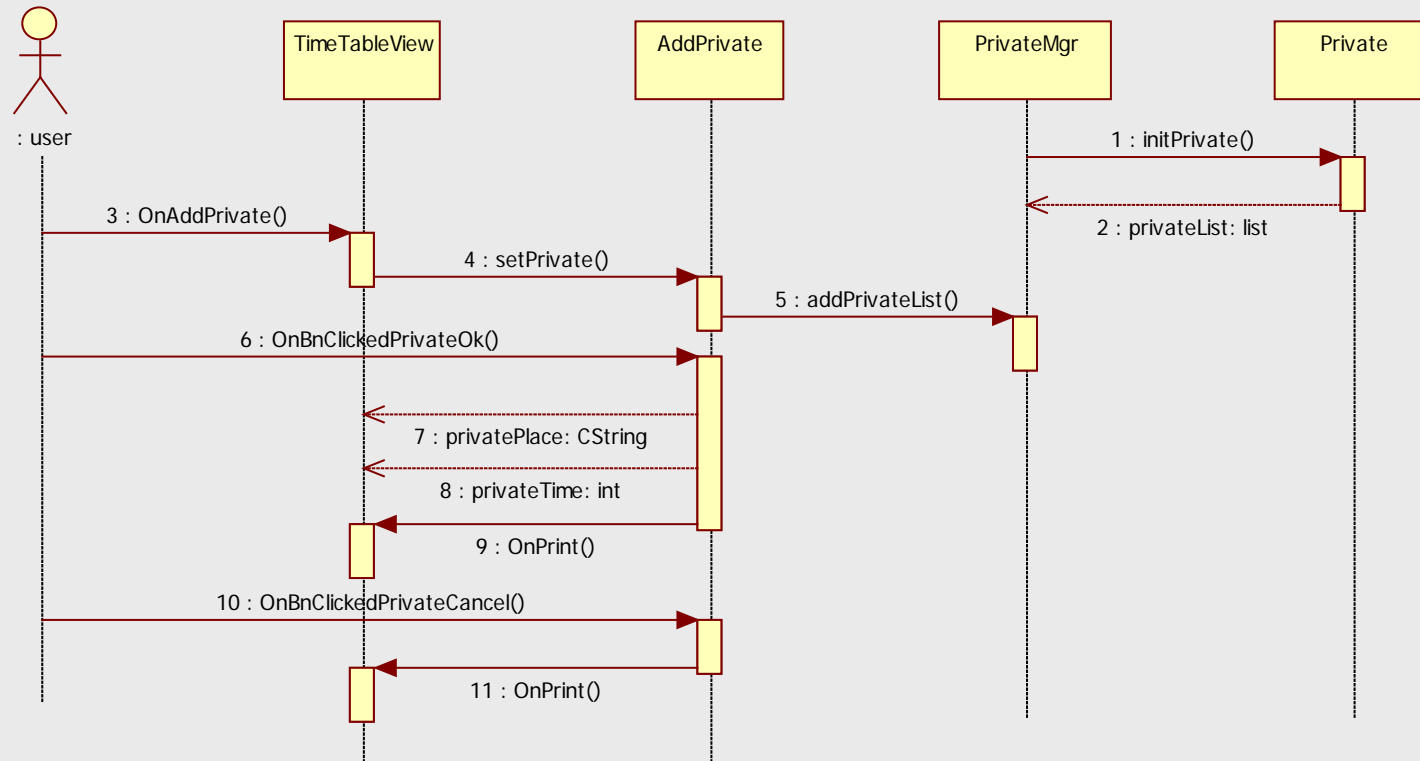
4. Interface description

- Class, ClassMgr



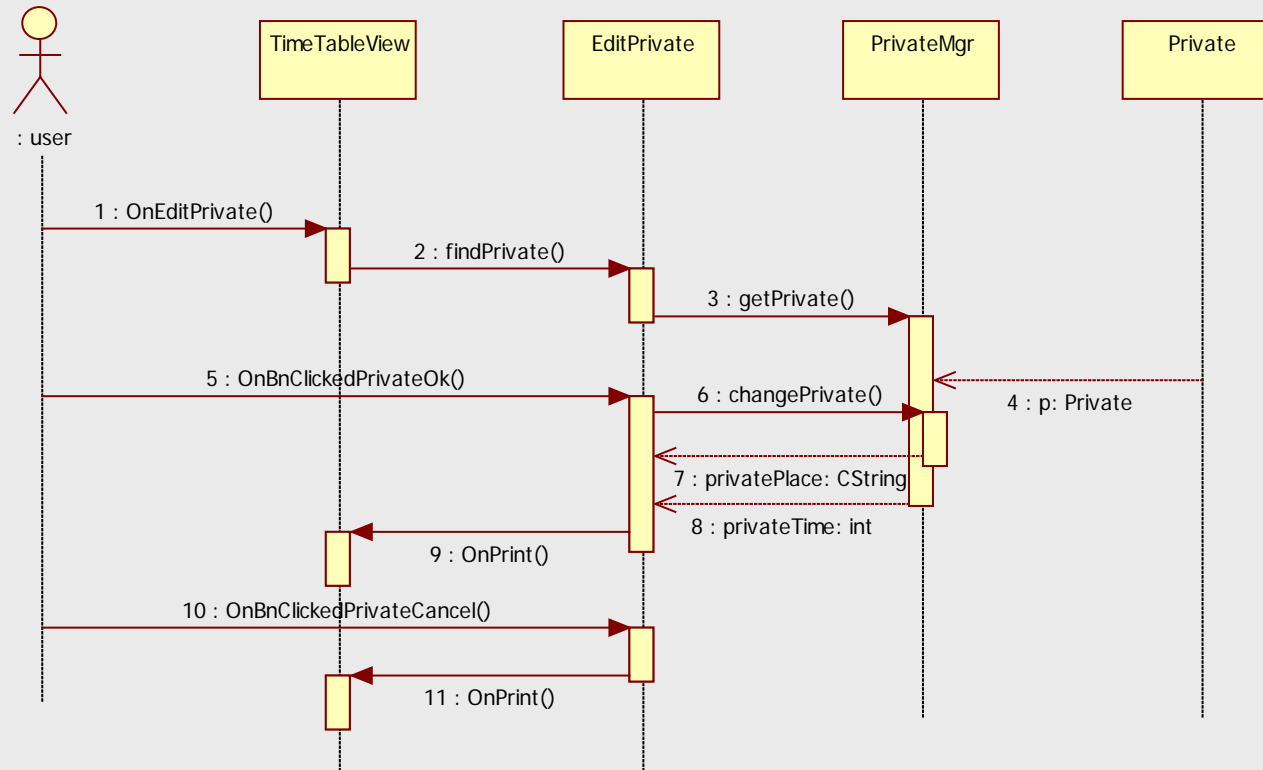
4. Interface description

- Private, PrivateMgr



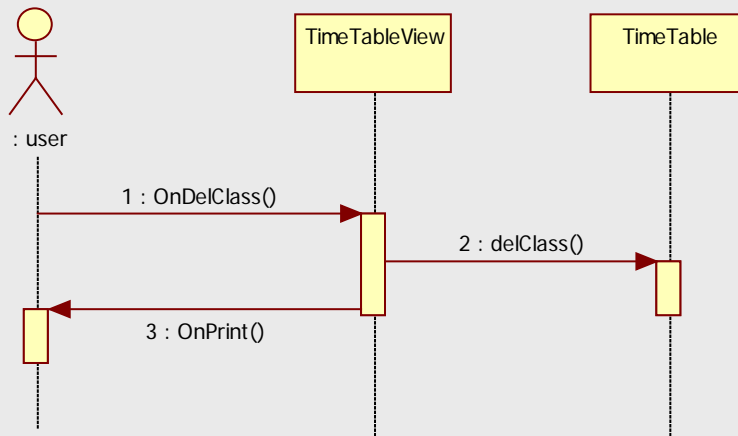
4. Interface description

- Edit Private

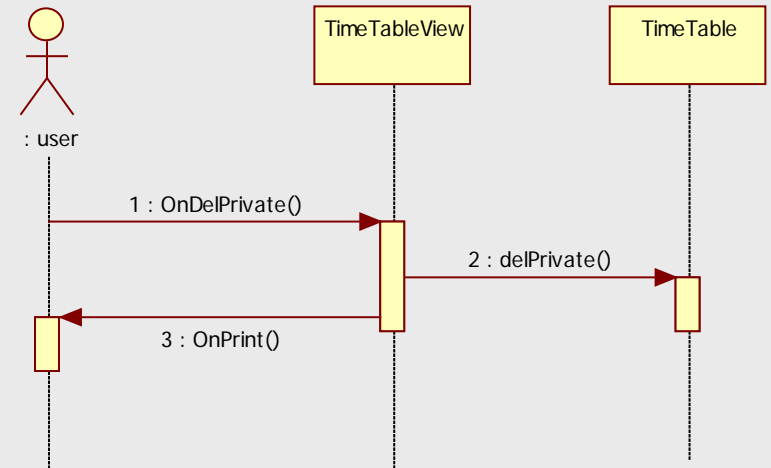


4. Interface description

- delClass, delePrivate



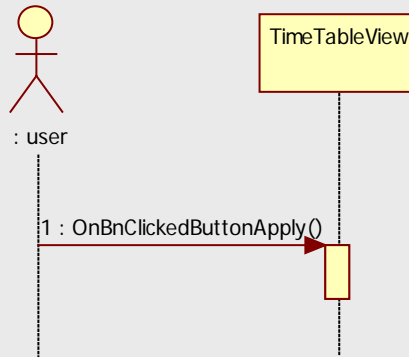
(1) delClass



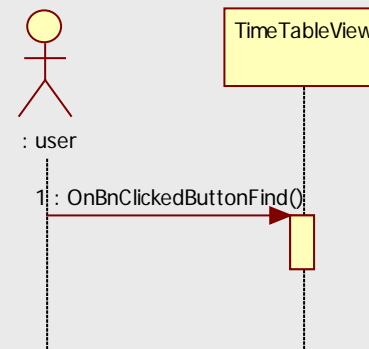
(2) delPrivate

4. Interface description

- show 수강신청, 종합강의시간표



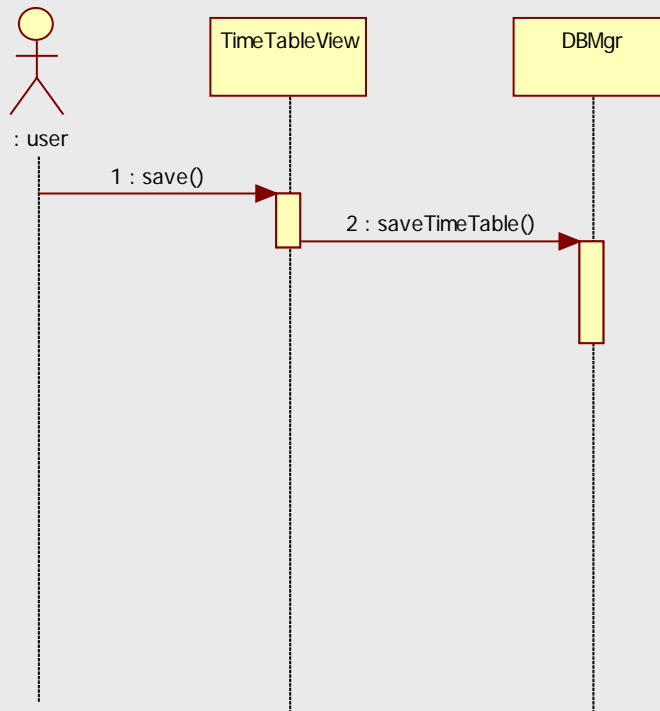
(1)수강신청



(2)종합강의시간표

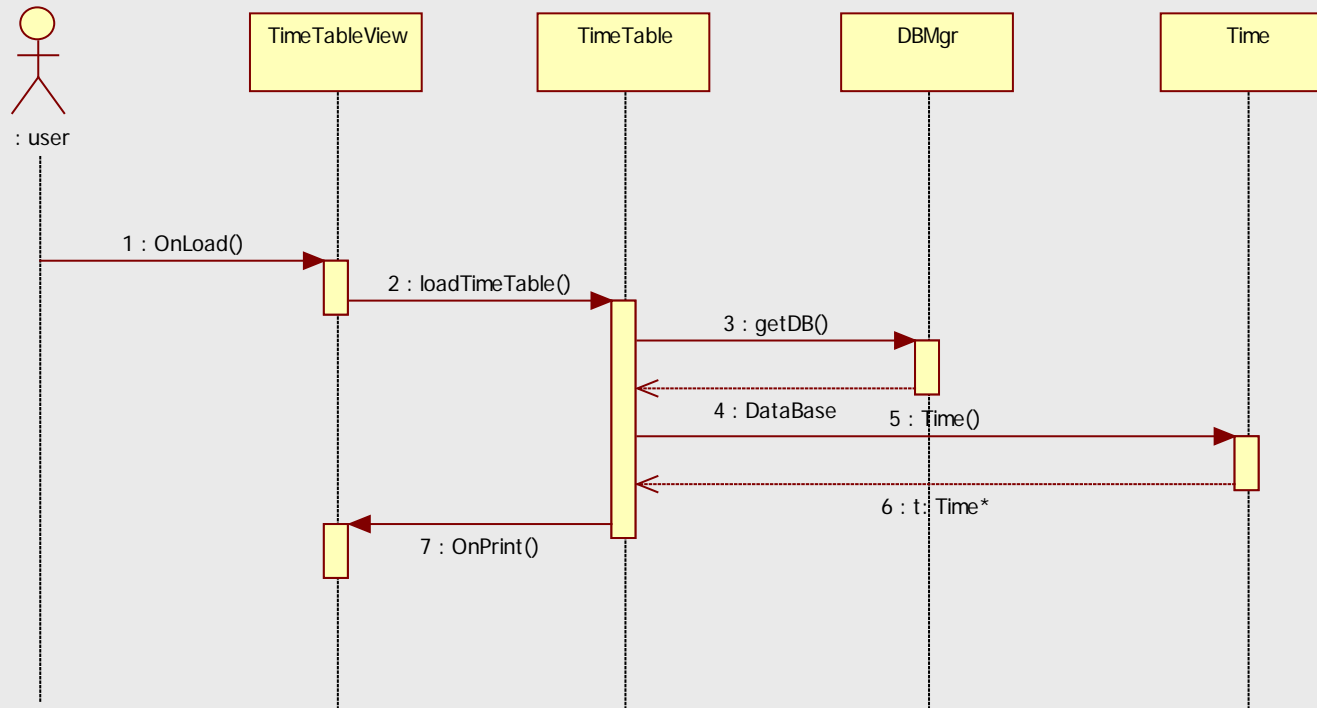
4. Interface description

- Save



4. Interface description

- Load



5. Detailed design description

Use Case

파란색 : method 후보

1. 사용자가 프로그램을 실행 한다
2. 사용자는 저장된 시간표를 불러오거나 새로이 작성(초기화)한다.
 - 2-1 사용자가 시간표를 새로이 작성한다.
 - 2-1-1 사용자는 등록하고 싶은 시간의 칸을 선택한다.(right click)
 - 2-1-1-1 사용자가 선택한 칸에 비어있다.
 - 2-1-1-1-1 '강의 등록' 을 선택한다.
 - 2-1-1-1-1-1 강의명, 교수명, 강의실, 시간을 메뉴에서 선택한다.(팝업창)
 - 2-1-1-1-1-2 입력을 다했으면 '확인' 버튼을 누른다.
 - 2-1-1-1-1-2-3 시간표에 강의를 추가된다.
 - 2-1-1-1-2 '일정 등록' 을 선택한다.
 - 2-1-1-1-2-1 활동명, 장소, 시간을 입력한다.(팝업창)
 - 2-1-1-1-2-2 입력을 다했으면 '확인' 버튼을 누른다.
 - 2-1-1-1-2-3 시간표에 일정이 추가된다.
 - 2-1-2 사용자가 선택한 칸에 일정이 있다.
 - 2-1-2-1 '변경' 을 선택한다.
 - 2-1-2-2 '삭제' 를 선택한다.
 - 2-1-2-3 '강의 정보' 를 선택한다.
 - 2-1-2-3-1 선택한 강의에 대한 강의 정보 창이 뜬다.(팝업창)
 - 2-2 사용자가 저장된 시간표를 불러온다.
 - 2-2-1 위의 내용과 동일한 작업 반복
3. 사용자가 시간표를 저장한다.
4. 사용자가 '수강 신청' 버튼을 클릭한다.
 - 4-1 건국대학교 수강신청 팝업창(인터넷 팝업창)이 뜬다.
5. 사용자가 '전체 강의 검색' 버튼을 클릭한다.
 - 5-1 건국대학교 종합강의시간표(인터넷 팝업창)가 뜬다.
6. 사용자가 시간표를 종료한다.

5. Detailed design description

- 행동 별 카테고리 정리

TimeTable

실행한다.
 저장한다.
 불러온다.
 강의 등록을 선택한다.
 일정 등록을 선택한다.
 변경을 선택한다.
 삭제를 선택한다.
 강의정보를 선택한다.
 화면에 출력한다.
 칸을 선택한다.

AddClass

강의명을 메뉴에서 선택한다.
 확인버튼을 누른다.
 취소버튼을 누른다.

Class

db로부터 data를 불러와
 변수에 대입한다.

AddPrivate

일정 이름, 시간을 입력한다.
 확인버튼을 누른다.
 취소 버튼을 누른다.

Private

db로부터 data를 불러와
 변수에 대입한다.

Time

일정이 추가된다.

5. Detailed design description

- Class 별 attribute, operation
 - Class

Class
+className: CString +classHour: int +classMinute: int +classTeacher: CString +classRoom: CString
+Class(name: CString, teacher: CString, room: CString, hour: int, minute: int)

5. Detailed design description

- Class 별 attribute, operation
 - ClassMgr, AddClass

ClassMgr
+classArray: Class[*]
+initClass(classArray: Class*) +getClass(classNum: int): c: Class +getClass(className: CString): c: Class

AddClass
+className: CString +classTeacher: CString +classRoom: CString +classHour: int +classMinute: int +classList: CComboBox +classNumList: CComboBox
+OnBnClickedAddCancel() +OnBnClickedAddOk() +findClass(classNum: int) +findClass(className: CString)

5. Detailed design description

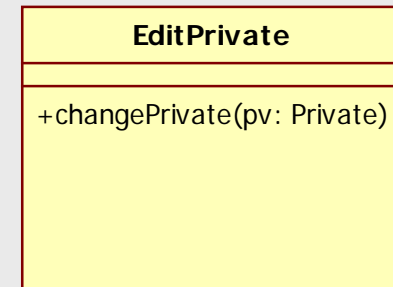
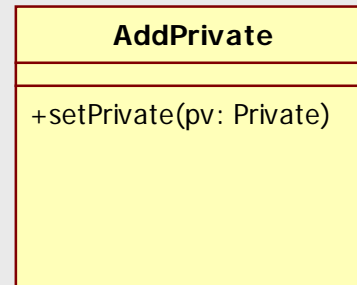
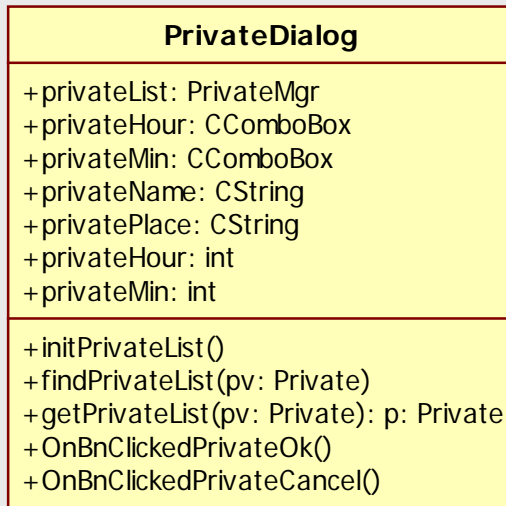
- Class 별 attribute, operation
 - Private, PrivateMgr

Private
+privateHour: int +privateMin: int +privateName: CString +privatePlace: CString
+initPrivate(hour: int, min: int, name: CString, place: CString)

PrivateMgr
+p: Private[*]
+initPrivateList() +addPrivateList(pv: Private) +findPrivate(pv: Private) +getPrivate(pv: Private): p: Private +delPrivate(pv: Private)

5. Detailed design description

- Class 별 attribute, operation
 - PrivateDialog, AddPrivate, EditPrivate



5. Detailed design description

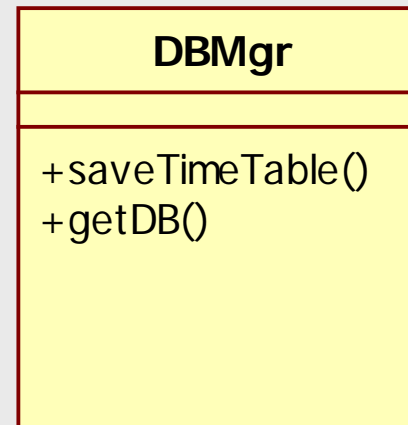
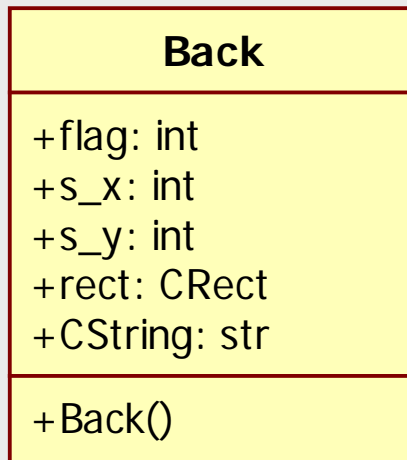
- Class 별 attribute, operation
 - Time, TimeTable

Time
+className: CString +classTime: int +classTeacher: int +classRoom: int
+Time() +Time(t: const Time&) +Time(name: CString, teacher: CString, room: CString, time: int) +Time(name: CString, room: CString, time: int) +operator=(t: const Time&)

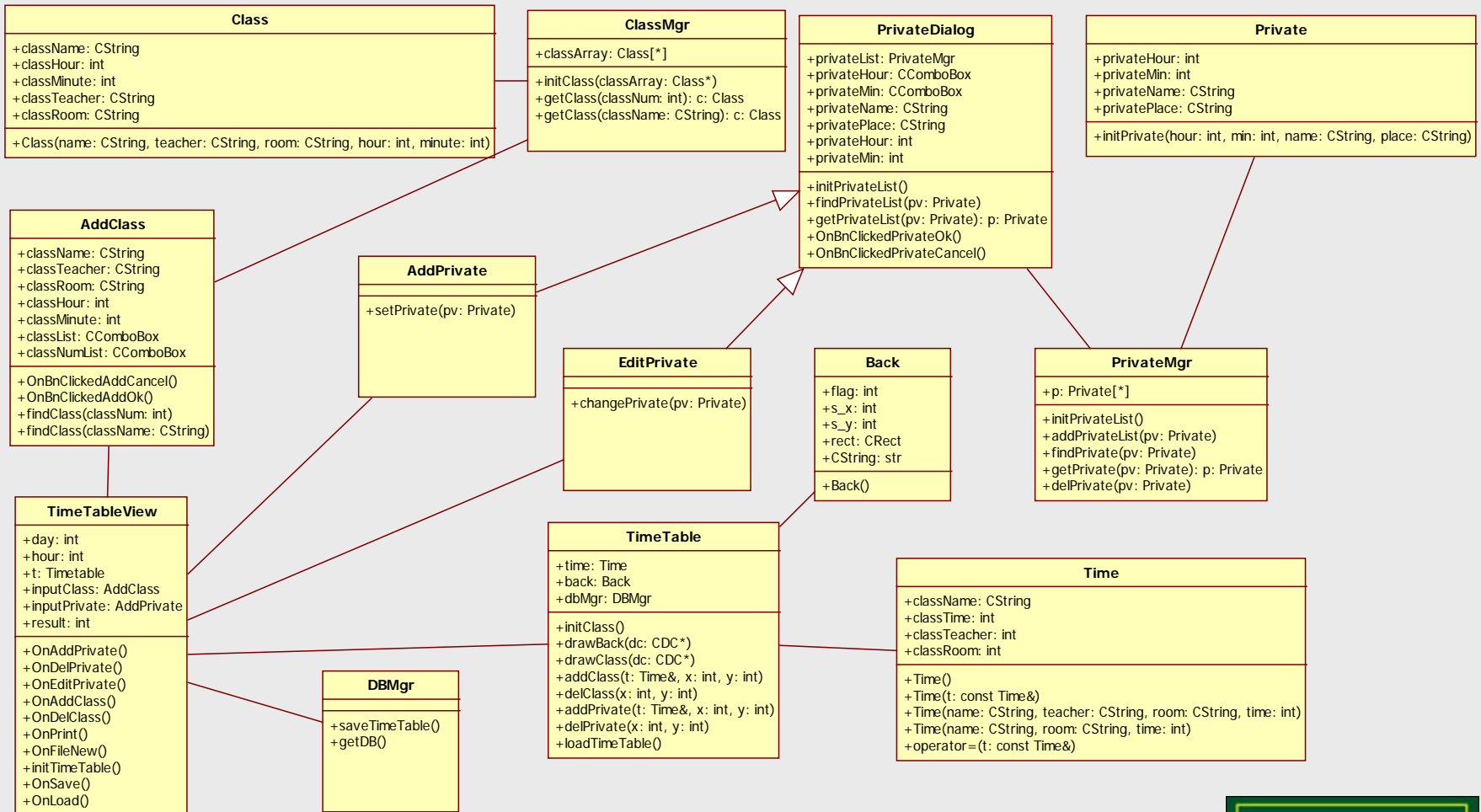
TimeTable
+time: Time +back: Back +dbMgr: DBMgr
+initClass() +drawBack(dc: CDC*) +drawClass(dc: CDC*) +addClass(t: Time&, x: int, y: int) +delClass(x: int, y: int) +addPrivate(t: Time&, x: int, y: int) +delPrivate(x: int, y: int) +loadTimeTable()

5. Detailed design description

- Class 별 attribute, operation
 - Back, DBMgr



5. Detailed design description



Conclusion

- IEEE SDD를 따라 class diagram을
 - I. decomposition description
 - II. dependency description
 - III. interface description
 - IV. detail description
- OOD 기법을 따라 설계

Reference

- Head first OOAD – 브렛 맥리프란 외 (O'REILLY)
- UML 객체지향 분석·설계 – 조완수 (홍릉과학출판사)
- Software Engineering (8th) – Ian Sommerville (Addison–Wesley)

End

감사합니다.