Perfect Time Table

Implementation & Demo

Team 8
200611458 김영승
200611478 성두훈
200611494 원스타
200611518 조민경

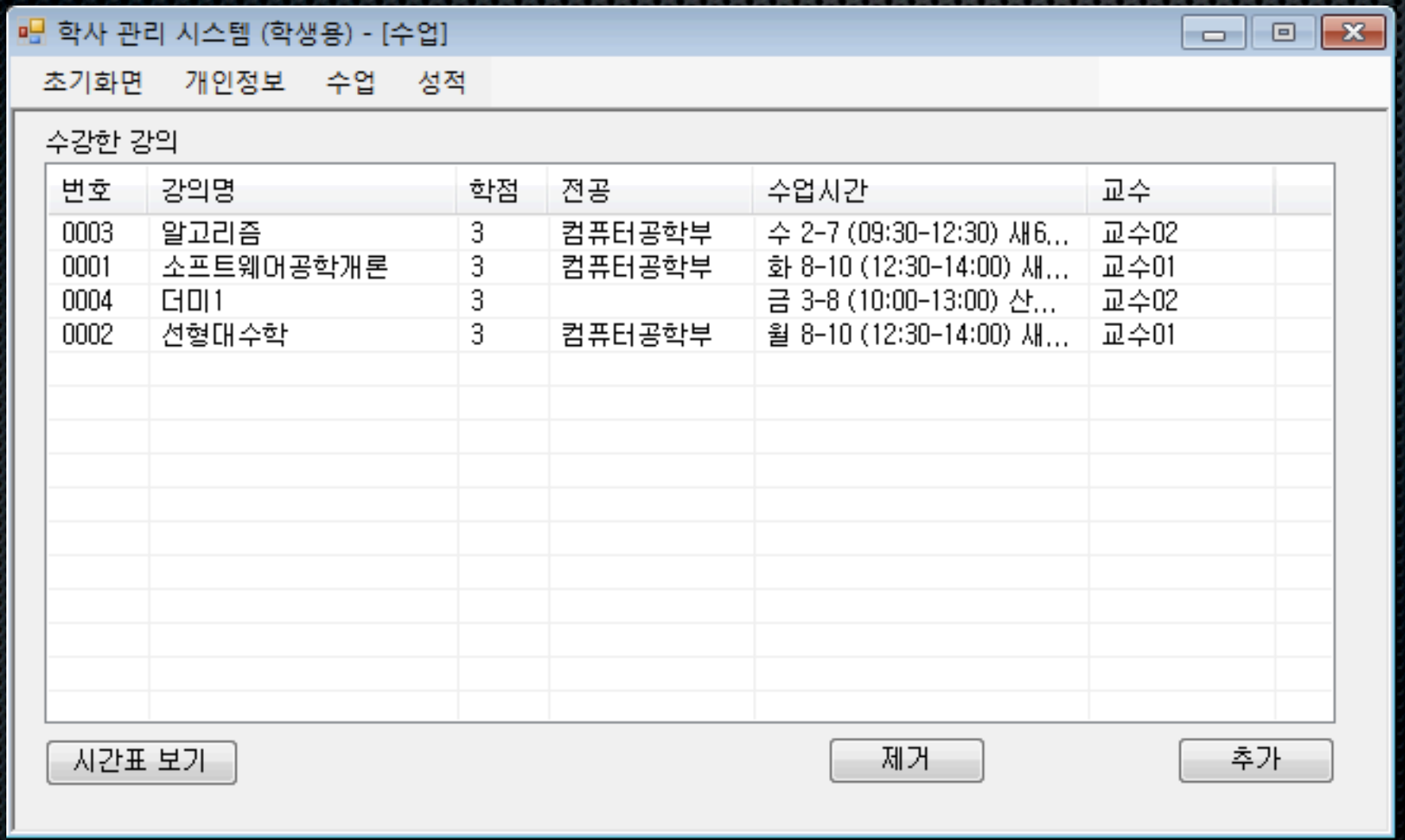
Contents

- Introduction
- UML Tool
- Implementation Details
- Demo

Introduction

Perfect Time Table

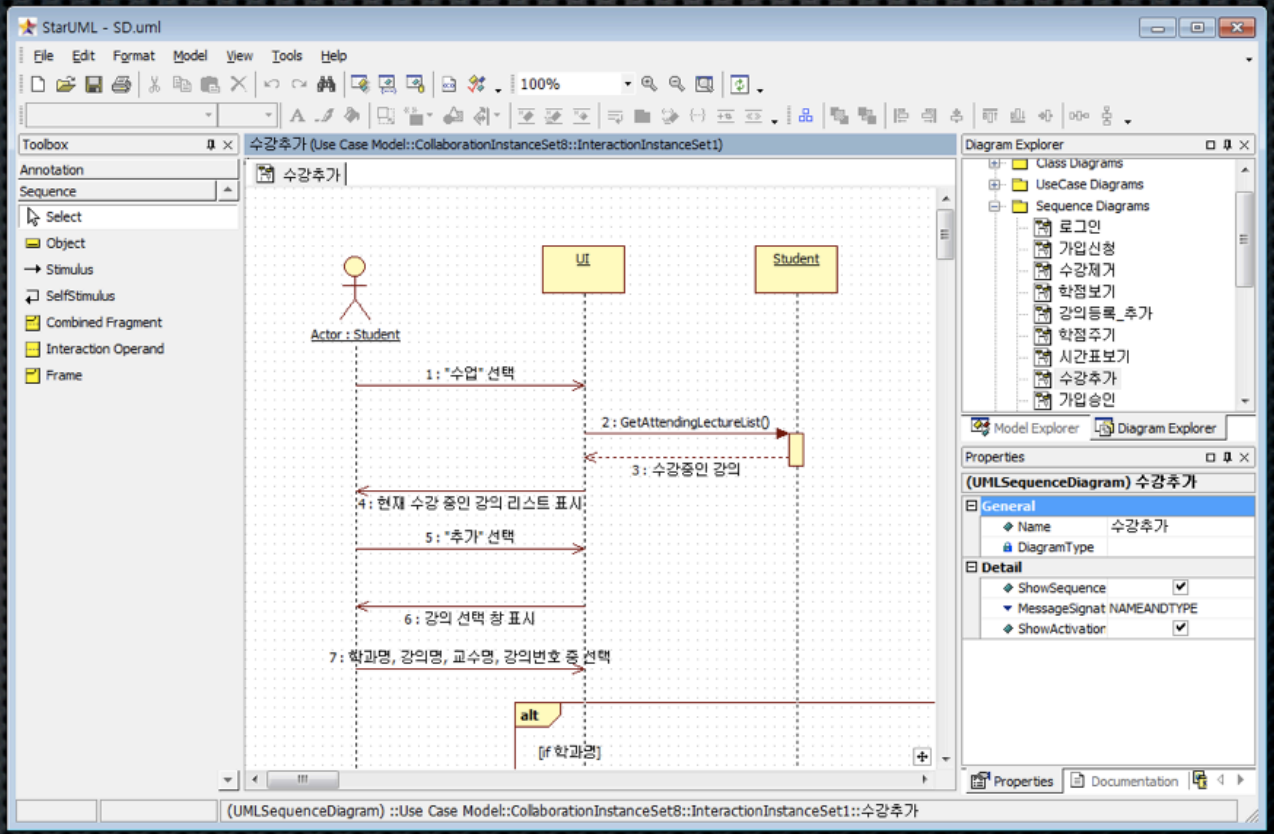
Perfect Time Table



UML Tool

UML Tool

- StarUML



UML Tool - Code Generation

- StarUML에서 C++/CLI를 지원하지 않음
- 일반 C++용 코드 생성 후 C++/CLI 스타일에 맞게 수정

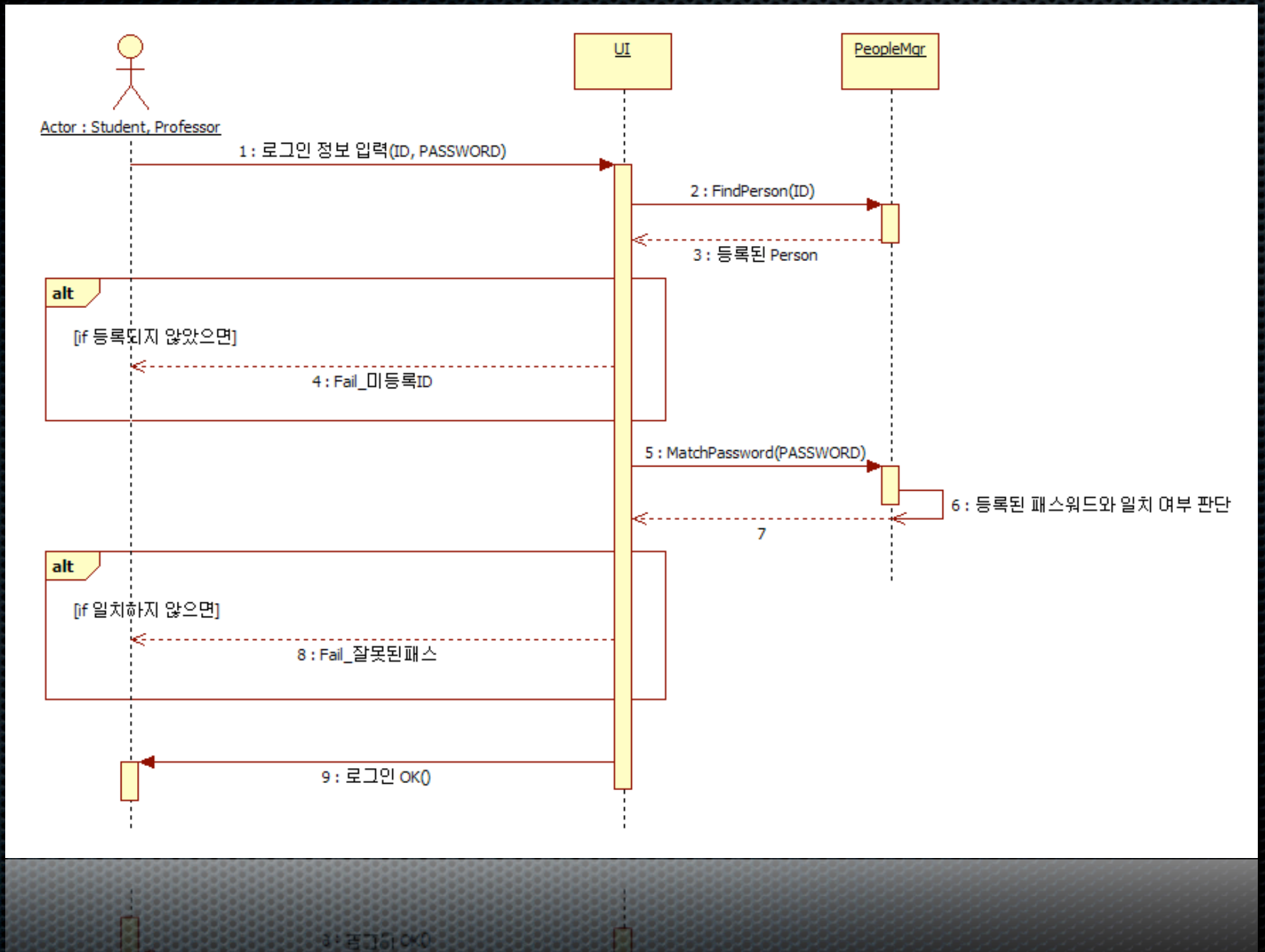
Implementation Details

Tracing Operations

- Operation: 사용자의 타입과 신원을 구분
- Requirements
  - SRS 3.1.1: 사용자별로 각기 다른 인터페이스를 제공해야 한다.
  - SRS 3.2.1: 프로그램 실행시 로그인 화면이 실행. 아이디, 패스워드 입력을 통해 로그인

Tracing Operations

- Design: SDD 6.2.3 Sequence Diagram "로그인"


Actor : Student, Professor
UI
PeopleMgr
1 : 로그인 정보 입력(ID, PASSWORD)
2 : FindPerson(ID)
3 : 등록된 Person
alt
[if 등록되지 않았으면]
4 : Fail_미등록ID
5 : MatchPassword(PASSWORD)
6 : 등록된 패스워드와 일치 여부 판단
7
alt
[if 일치하지 않으면]
8 : Fail_잘못된패스
9 : 로그인 OK()

Tracing Operations

- Implementation: class LoginForm

Tracing Operations

TimeTable::LoginForm | LoginProc(System::String ^ userId, System::String ^ password)

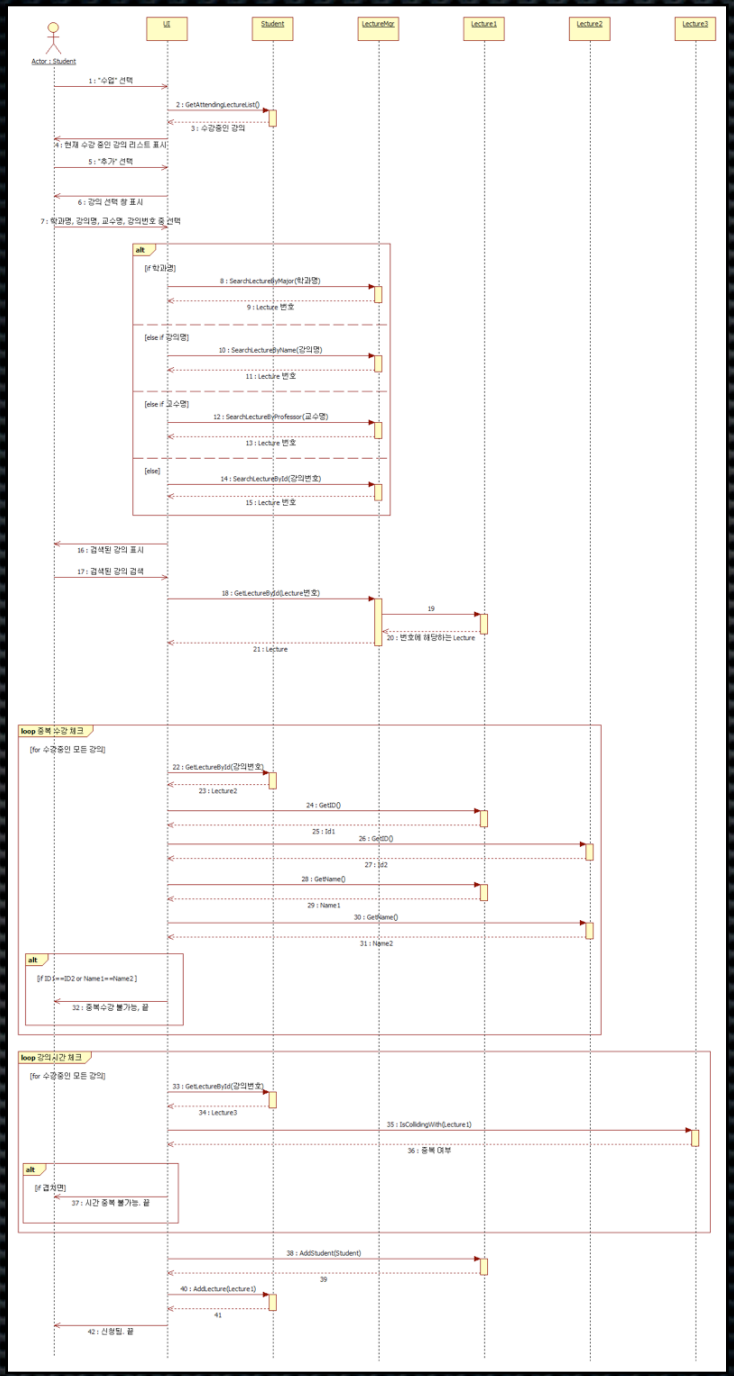
```
82          //
83      }
84
85      LoginForm::LoginResult LoginForm::LoginProc(String^ userId, String^ password)
86      {
87          Person^ pFoundPerson = m_pPeopleMgr->FindPerson(userId);
88          if (!pFoundPerson)
89          {
90              return LoginResult::kLoginFail_UnknownId;
91          }
92
93          if (!pFoundPerson->MatchPassword(password))
94          {
95              return LoginResult::kLoginFail_IncorrectPassword;
96          }
97
98          m_pLoggedPerson = pFoundPerson;
99          // 로그인 처리
100         return LoginResult::kLoginOk;
101     }
102
103     LoginForm::LoginResult LoginForm::GetLoginResult()
104     {
105         return m_loginResult;
106     }
107
```

Tracing Operations

- Operation: 학생 -> 강의 수강 신청
- Requirements: SRS 3.2.3.a "학생 -> 수업관리" 참조

Tracing Operations

- Design: SDD 6.2.6 Sequence Diagram "수강신청"


Actor : Student
UI
Student
LectureMgr
Lecture1
Lecture2
Lecture3
1 : "수업" 선택
2 : GetAttendingLectureList()
3 : 수강중인 강의
4 : 현재 수강 중인 강의 리스트 표시
5 : "추가" 선택
6 : 강의 선택 창 표시
7 : 학과명, 강의명, 교수명, 강의번호 중 선택
alt
[if 학과명]
8 : SearchLectureByMajor(학과명)
9 : Lecture 번호
[else if 강의명]
10 : SearchLectureByName(강의명)
11 : Lecture 번호
[else if 교수명]
12 : SearchLectureByProfessor(교수명)
13 : Lecture 번호
[else]
14 : SearchLectureById(강의번호)
15 : Lecture 번호
16 : 검색된 강의 표시
17 : 검색된 강의 검색
18 : GetLectureById(Lecture번호)
19
20 : 번호에 해당하는 Lecture
21 : Lecture
loop 중복 수강 체크
[for 수강중인 모든 강의]
22 : GetLectureById(강의번호)
23 : Lecture2
24 : GetID()
25 : Id1
26 : GetID()
27 : Id2
28 : GetName()
29 : Name1
30 : GetName()
31 : Name2
alt
[if ID1==ID2 or Name1==Name2]
32 : 중복수강 불가능, 끝
loop 강의시간 체크
[for 수강중인 모든 강의]
33 : GetLectureById(강의번호)
34 : Lecture3
35 : IsCollidingWith(Lecture1)
36 : 중복 여부
alt
[if 겹쳐면]
37 : 시간 중복 불가능. 끝
38 : AddStudent(Student)
39
40 : AddLecture(Lecture1)
41
42 : 신청됨. 끝

Tracing Operations

- Implementation
  - class StudentAttendForm
  - class LectureListForm

Tracing Operations

- Operation: 학생 -> 성적 확인
- Requirements
  - SRS 3.2.4.a: “학생 -> 성적 및 강의 평가” 참조
- Design
  - SDD 6.2.10 Sequence Diagram "학점 보기"
- Implementation
  - class LectureGradeForm

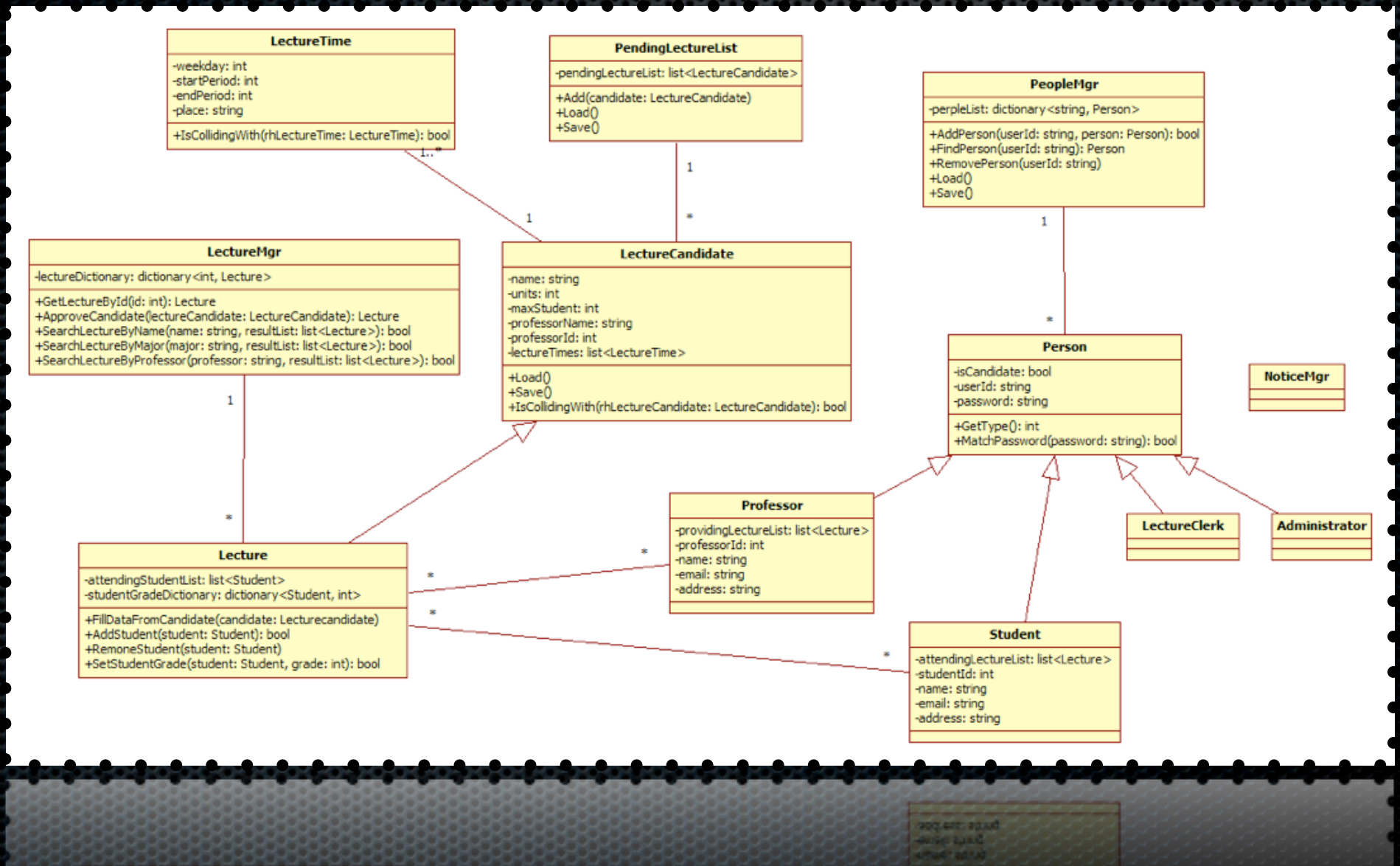
Tracing Operations

- Operation: 교수 -> 강의 등록 신청
- Requirements
  - SRS 3.2.6: “교수 -> 강의등록 신청” 참조
- Design
  - SDD 6.2.5 Sequence Diagram "강의등록_추가"
- Implementation
  - class ProfessorAddLectureForm

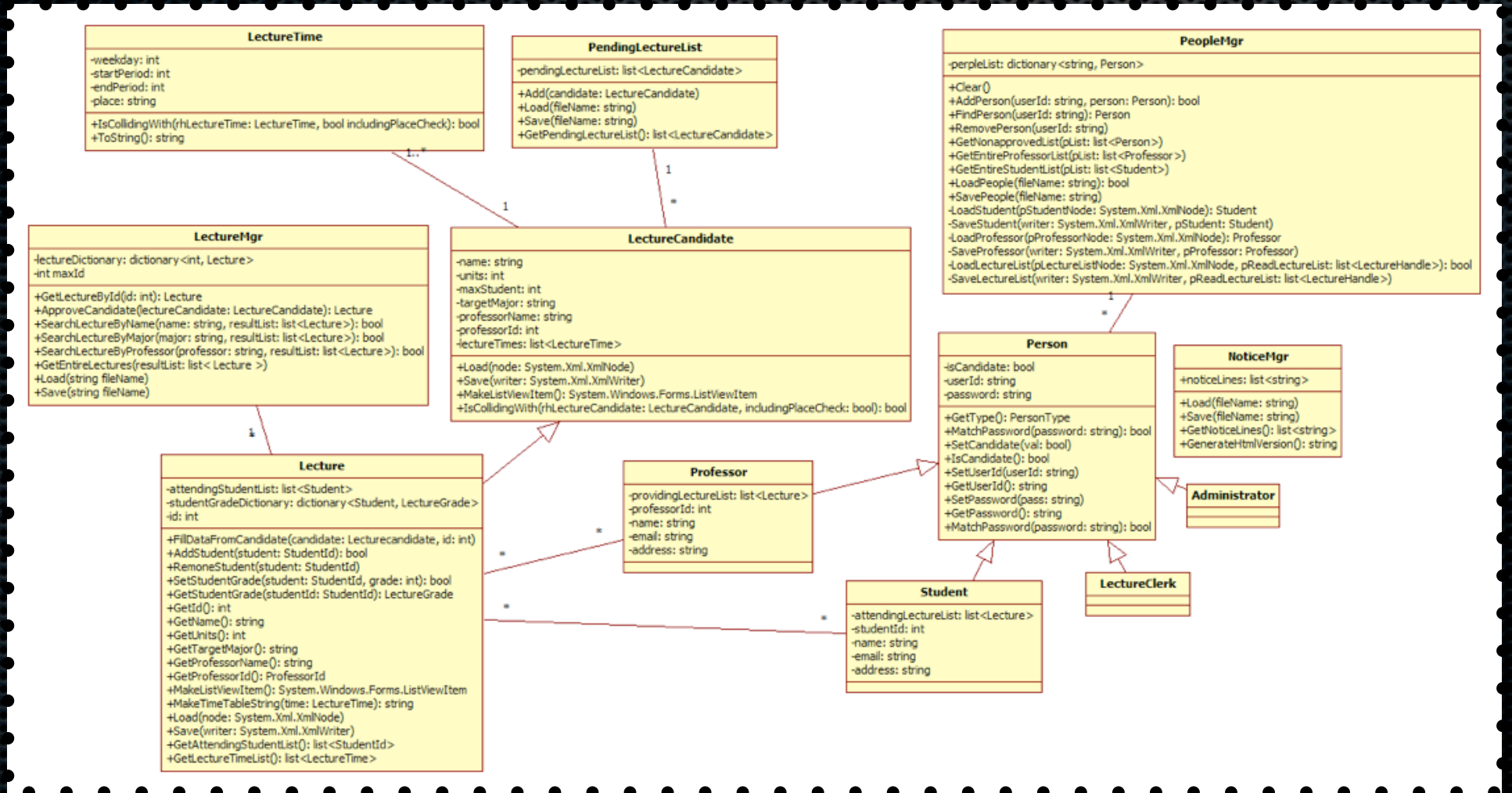
클래스 구조

- 큰 구조 변경은 없었음.
- 하지만 몇 가지 사소한(?) 변경점

디자인 단계의 클래스 다이어그램



구현 후 클래스 다이어그램



클래스 구조 변경의 이유

- UI 연동을 위해
- 명확한 자료형 구분
- 안정성을 위해

UI 연동

- UI 연동 작업에서 일어난 변동사항

- Getter, Setter 등
- MakeListViewItem()
- MakeTimeTableString()
- LectureTime::ToString()
- NoticeMgr class

명확한 자료형 구분

- int -> PersonType
- int -> ProfessorId
- int -> StudentId
- int -> LectureHandle

안정성

- 실제 객체의 포인터를 직접 참조할 경우
  무효화된 포인터로 인해 크래시 발생 가능성

- Handle 개념 도입

안정성

- attendingStudentList: List<Student>
- -> attendingStudentList: List<StudentId>
- providingLectureList: List<Lecture>
- -> providingLectureList: List<LectureHandle>
- attendingLectureList : List<Lecture>
- -> attendingLectureList : List<LectureHandle>

Demo

Any questions?

Thank you!