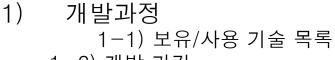


목차



- 1-2) 개발 기간
- 1-3) DB테이블
- 1-4) 순서도(기능분담)
- 1-5) UML

2) 기술설명

- 2-1) 민희원 회원관리 구현
- 2-2) 이혜련 놀이기구 관리 구현
- 2-3) 정병수 놀이기구 예약 구현
- 2-4) 임정통 게시판 구현

1-1)보유기술





















1-2] 개발기간

	단계	8/19	8/20	8/21	8/22
기획	프로젝트 기획				
설계	데이터베이스 설계				
코딩	레이아웃 작업				
	코딩				
구현	디버깅				

	단계	8/23	8/24	8/25	8/26
기획	프로젝트 기획				
설계	데이터베이스 설 계				
코딩	레이아웃				
	코딩				
구현	디버깅				

1-3) DB테이블

Members table

속성	속성타입	제약조건	내용
M_id	varchar(100)	Not null, primary	회원 아이디
Pw	varchar(100)	Not null	비밀번호
M_name	varchar(100)		회원 이름
Gender	varchar(100)		성별 (M or F)
Birth	varchar(100)		출생년도
Tall	Int(11)	Not null	키 (cm)
Tel	varchar(100)		전화번호
Email	varchar(100)		이메일

1-3) DB테이블

Board table

속성	속성타입	제약조건	내용
B_id	Int(11)	Not null, primary	게시글 아이디
M_id	varchar(100)	Foreign key, references Members	회원 아이디
title	varchar(100)		제목
content	text		내용
uDate	datetime		게시일시
hits	Int(11)		조회수

1-3) DB테이블

Ride table

속성	속성타입	제약조건	내용
R_id	varchar(100)	Not null, primary	놀이기구 아이디
R_name	varchar(100)		놀이기구 이름
Location	varchar(100)		놀이기구 위치
people	Int(11)		수용인원
R_Ltall	Int(11)		탑승가능 최소 키
R_Htall	Int(11)		탑승가능 최대 키
ITime	Int(11)		놀이기구 소요 시간
R_content	text		놀이기구 내용
Availability	varchar(100)	Not null	예약가능 여부
Img	varchar(100)		놀이기구 사진

1-3) DB테이블

1)개발과정 Ptable table

속성	속성타입	제약조건	내용
P_id	varchar(100)	Not null, primary	입장권 아이디
P_name	varchar(100)		입장권 종류 이름
price	Int(11)		가격

ttable table

속성	속성타입	제약조건	내용
T_id	varchar(100)	Not null	타임테이블 아이디
R_id	varchar(100)	Foreign key, references Ride	놀이기구 아이디
R_name	varchar(100)		놀이기구 이름
sTime	varchar(100)		시작시간
fTime	varchar(100)		종료시간
bTime	varchar(100)		탑승시간

1-3) DB테이블

SmartR table

속성	속성타입	제약조건	내용
SR_id	varchar(100)	Not null, primary	상품예약번호
P_id	varchar(100)	Foreign key, references Ptable	상품번호
P_name	varchar(100)		상품명
rDate	varchar(100)		예약날짜
price	Int(11)		상품가격
M_id	varchar(100)	Foreign key, references Members	회원 아이디
M_name	varchar(100)		회원 이름
kdate	datetime		데이터 적재시간

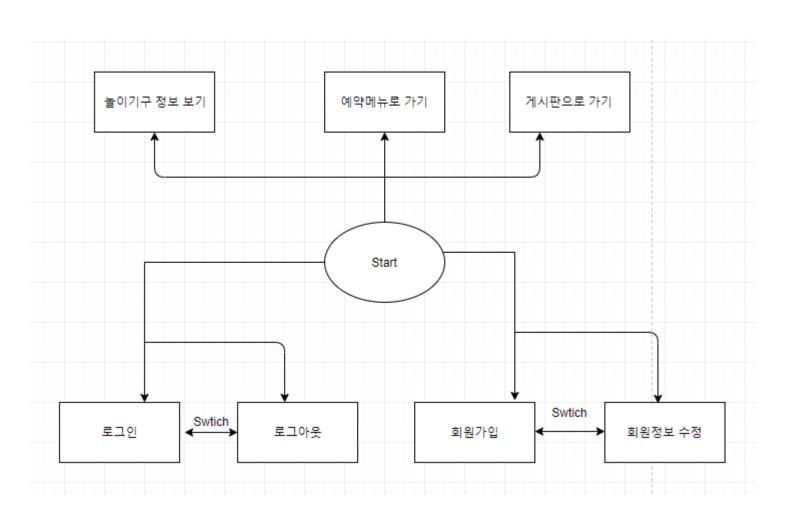
1-3) DB테이블

Rider table

Tilder tak	710		
속성	속성타입	제약조건	내용
Rr_id	varchar(100)	Not null, primary	놀이기구 예약번호
R_name	varchar(100)		놀이기구 이름
Sr_id	varchar(100)	Foreign key, references Smartr	예약번호
rdate	varchar(100)		탑승날짜
sTime	varchar(100)		회차 시작시간
fTime	varchar(100)		회차 종료시간
bTime	varchar(100)		탑승시간
M_id	varchar(100)	Foreign key, references Members	회원 아이디
M_name	varchar(100)		회원 이름
R_id	varchar(100)	Foreign key, references Ride	놀이기구 아이디
T_id	varchar(100)	Foreign key, references ttable	타임테이블 아이디
kdate	datetime		데이터 적재시간

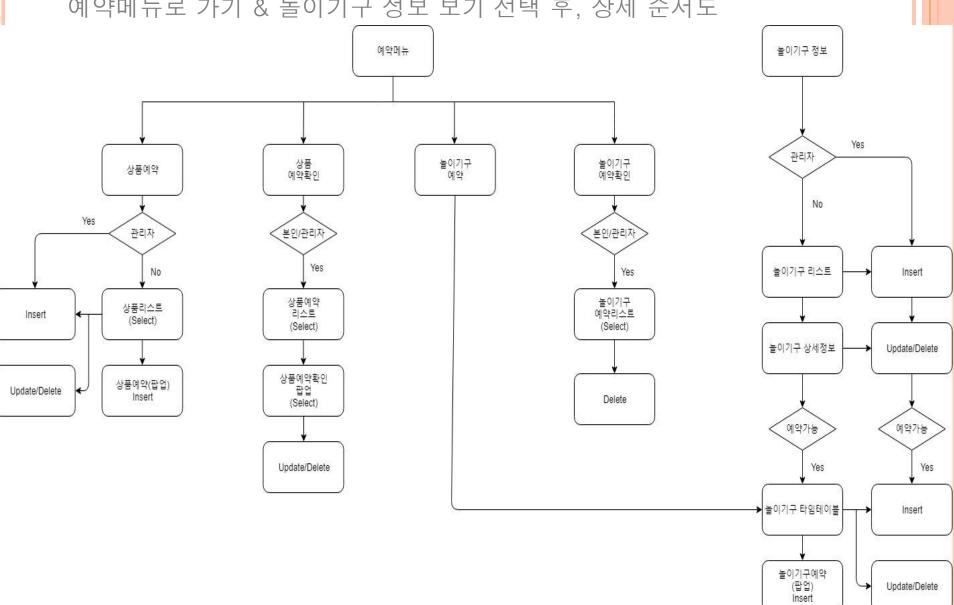
1-4) 순서도

첫 화면 메인 순서도



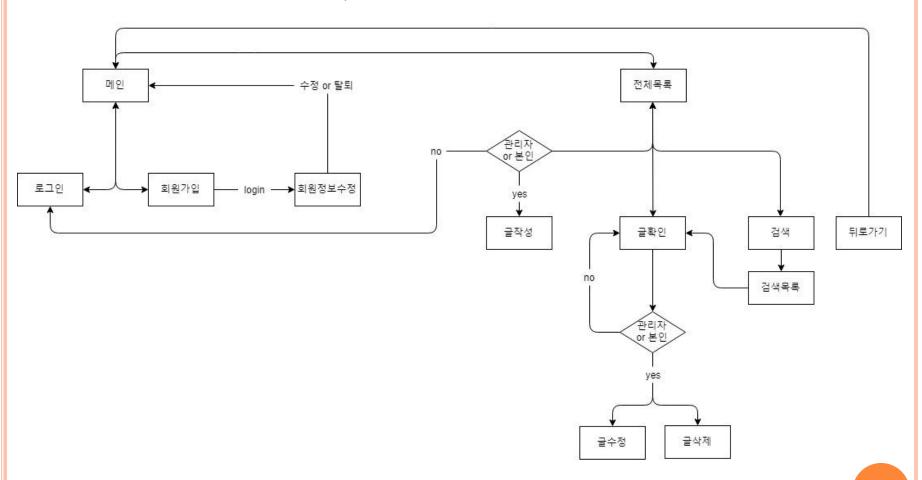
1-4) 순서도

예약메뉴로 가기 & 놀이기구 정보 보기 선택 후, 상세 순서도



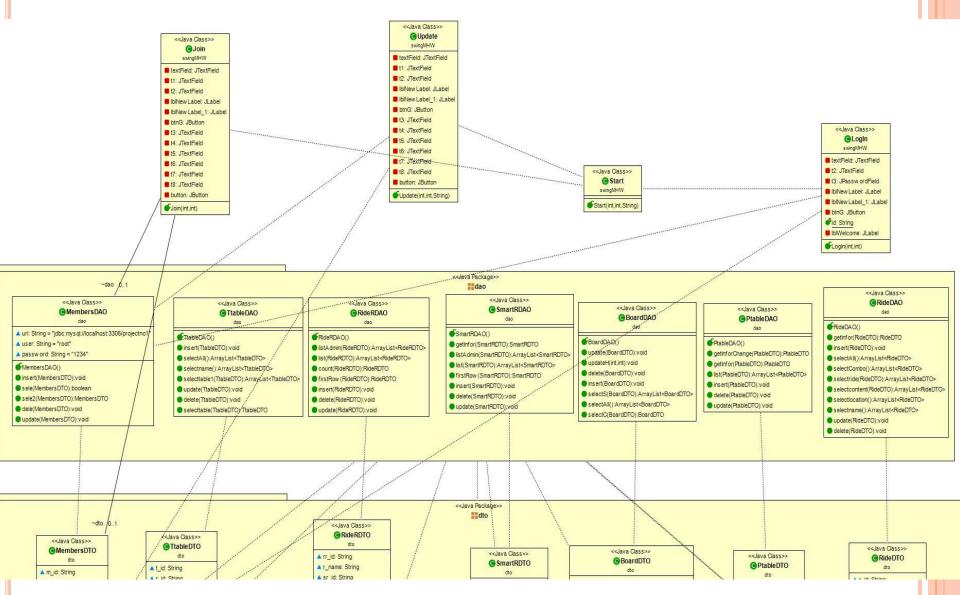
1-4) 순서도

게시판으로 가기 선택 후, 상세 순서도



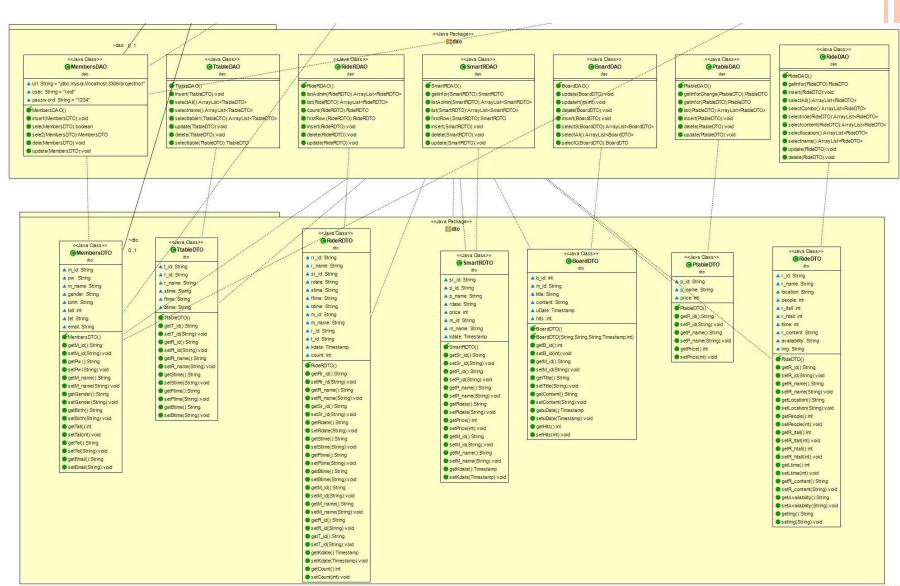
1-5) UML

(1) SwingMHW, Join, Update, Login



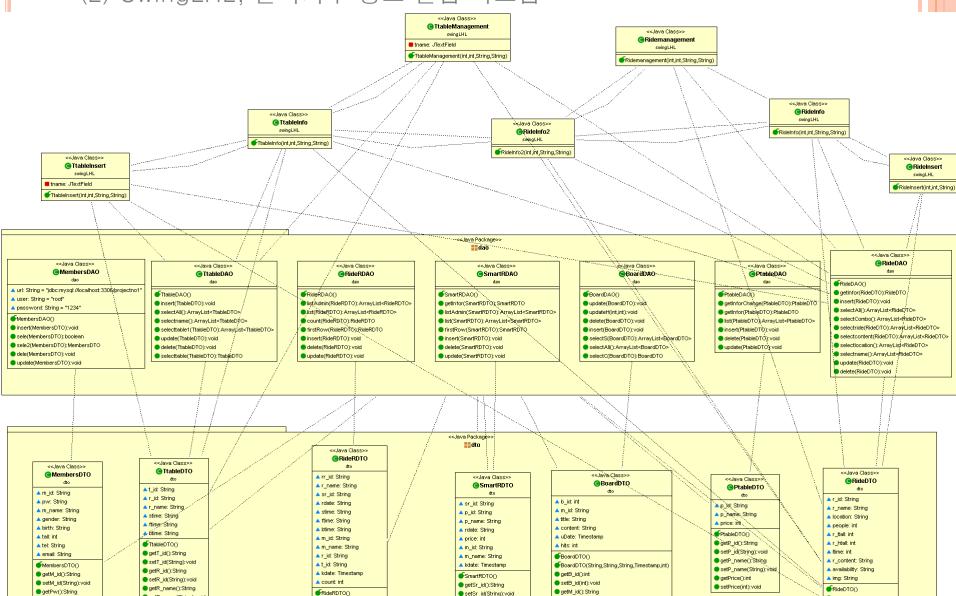
1-5) UML

(1) SwingMHW, Join, Update, Login - DTO, DAO 연결상태



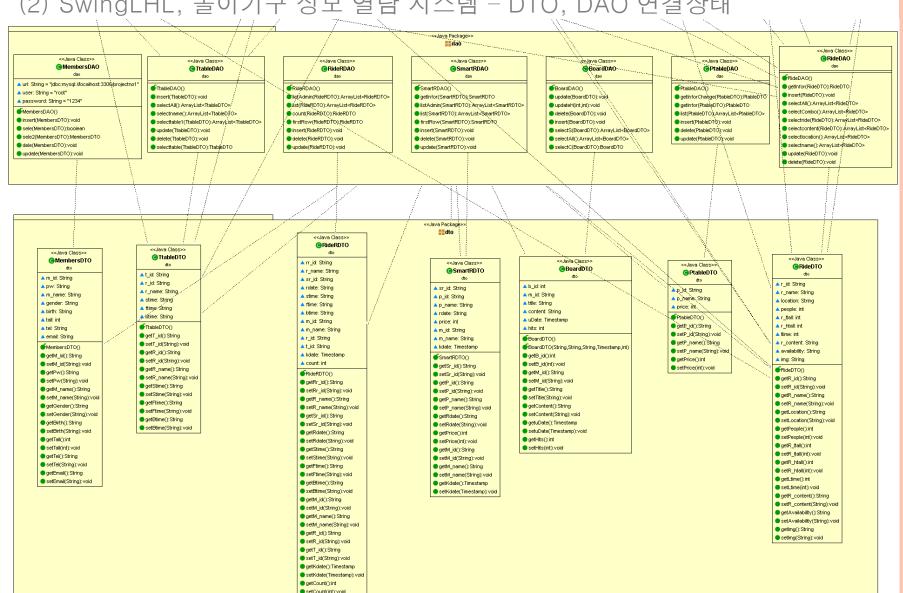
1-5) UML

(2) SwingLHL, 놀이기구 정보 열람 시스템



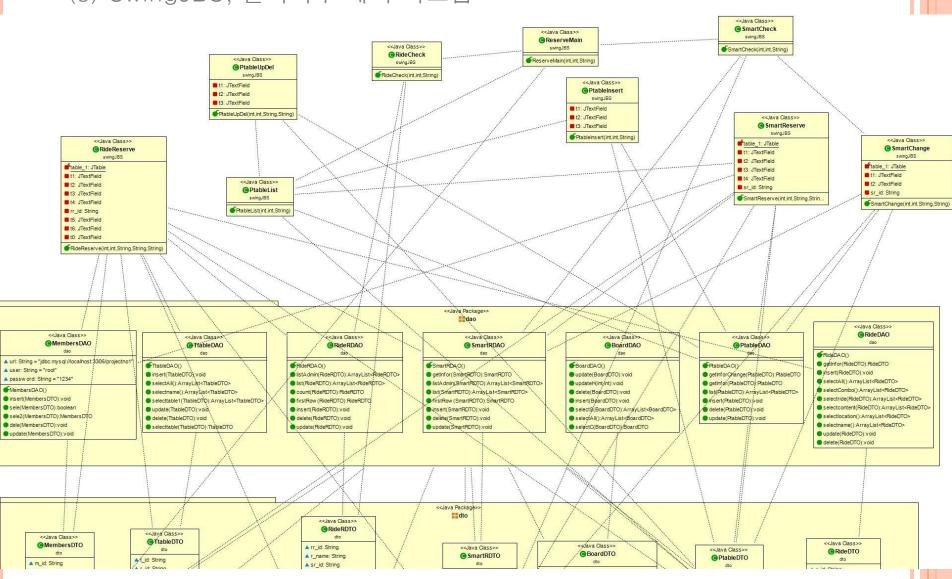
1-5) UML

(2) SwingLHL, 놀이기구 정보 열람 시스템 - DTO, DAO 연결상태



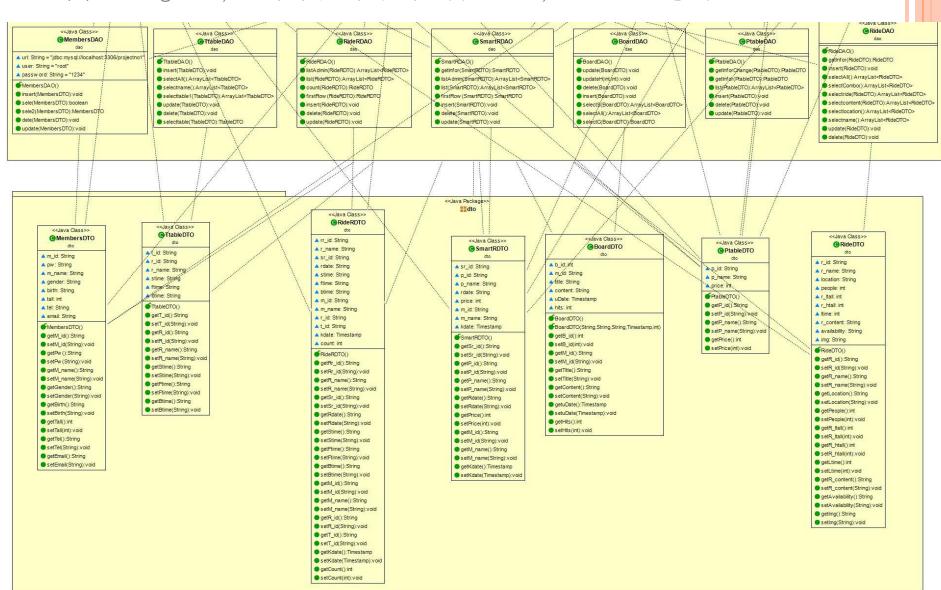
1-5) UML

(3) SwingJBS, 놀이기구 예약 시스템



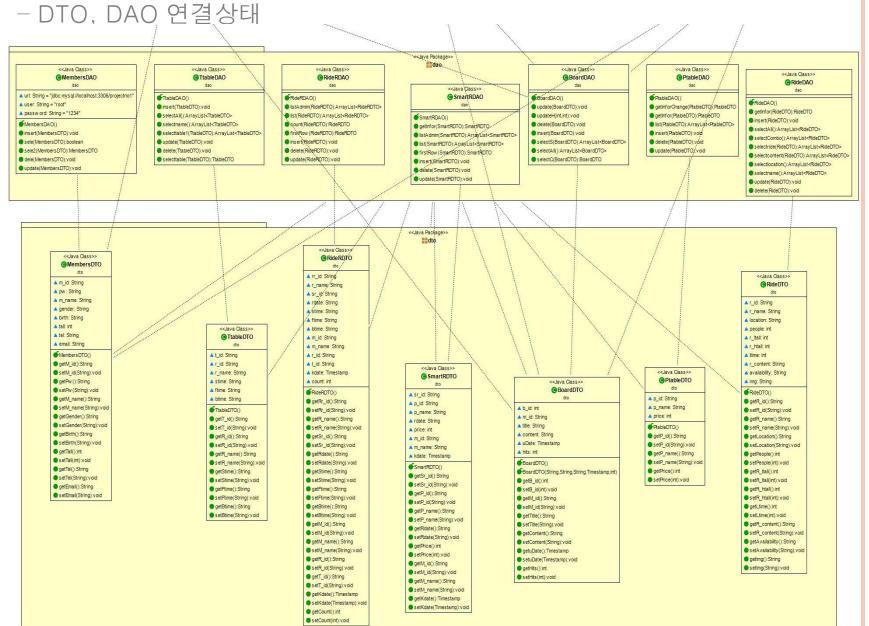
1-5) UML

(3) SwingJBS, 놀이기구 예약 시스템-DTO, DAO 연결상태



1]개발과정 (4) SwingLJT, 게시판 1-51 UMI <<Java Class>> <<Java Class>> BoardWrite BoardMain BoardConfirm swingIJT swingIJT swingIJT t1: JTextField t0: JTextField t1: JTextField t2: JTextField ▲ f: JFrame t2: JTextField BoardWrite(int,int,String) ◆BoardMain(int,int,String) ■BoardConfirm(int,int,String,int) <<Java Package>> <<Java Class>> <<Java Class>> <<Java Class>> <<Java Class>> <<Java Class>> <<Java Class>> Members DAO TtableDAO RideRDAO **@**BoardDAO PtableDAO @RideDAO <<Java Class>> ▲ url: String = "jdbc:mysql://localhost:3306/projectno1" FRideRDAO() **⊚** SmartRDAO BoardDAO() FrableDAO() FrideDAO() ▲ user: String = "root" update(BoardDTO):void getinforChange(PtableDTQ):PtableDTQ insert(TtableDTO):void listAdmin(RideRDTO):ArrayList<RideRDTO</p> getInfor(RideDTO):RideDTO A passw ord: String = "1234" selectAll():ArrayList<TtableDTO> list(RideRDTO):ArrayList<RideRDTO> updateH(int,int):void getinfor(PtableDTO):PtableDTO SmartRDAO() insert(RideDTO):void MembersDAO() selectname():ArrayList<TtableDTO> count(RideRDTO):RideRDTO delete(BoardDTO):void list(PtableDTO):ArrayList<PtableDTO> getInfor(SmartRDTO):SmartRDTQ selectAll():ArrayList<RideDTO> insert(Members DTO):void selecttable1(TtableDTO):ArrayList<TtableDTO> firstRow (RideRDTO):RideRDTO insert(BoardDTO):void insert(PtableDTO):void listAdmin(SmartRDTO):ArrayList<SmartRDTO> selectCombo():ArrayList<RideDTO> sele(MembersDTO):boolean update(TtableDTO):void insert(RideRDTO):void selectS(BoardDTO):ArrayList<BoardDTO;</p> delete(PtableDTO):void list(SmartRDTO):ArrayList<SmartRDTO> selectride(RideDTO):ArrayList<RideDTO> sele2(MembersDTO):MembersDTO delete(TtableDTO):void delete(RideRDTO):void selectAll():ArrayList<BoardDTO> update(PtableDTO) void firstRow (SmartRDTO):SmartRDTO selectcontent(RideDTO):ArravList<RideDTO> dele(MembersDTO):void selecttable(TtableDTO):TtableDTO update(RideRDTO):void selectC(BoardDTO):BoardDTO insert(SmartRDTO):void selectlocation():ArrayList<RideDTO> update(MembersDTO):void delete(SmartRDTO):void selectname():ArrayList<RideDTO> update(SmartRDTO):void update(RideDTO):void delete(RideDTO):void <<Java Package>> dto <<Java Class>> <<Java Class>> @RideRDTO MembersDTO dto ▲ rr_id: String <<Java Class>> ▲ m_id: String RideDTO ▲ r_name, String A pw : String ▲ sr_id/ String dto ▲ m_name: String ▲ rdate: String ▲ r_id: String ▲ gender: String stime: String ▲ r_name: String ▲ birth: String ftime: String ▲ location: String ▲ tall: int <<Java Class>> ▲ btime: String A people: int ▲ tel: String TtableDTO ▲ m_id: String ▲ r_Itall: int ▲ email: String ▲ m_name: String ▲ r_htall: int At id: String ▲ r id: String ▲ Itime: int getM_id():String ▲ r_id: String ▲ t id: String ▲ r_content: String << lava Class>> setM_id(String):void ▲ r_name: String ▲ kdate: Timestamp <<Java Class>> ▲ availability: String SmartRDTO OPtable DTO getPw ():String ▲ stime: String a count: int <<Java Class>> ▲ img: String dto setPw (String):void ▲ ftime: String BoardDTO FideRDTO() FrideDTO() ▲ sr_id: String getM_name():String ▲ btime: String ▲ p_id: String getRr_id():String getR_id():String ▲ p_id: String setM_name(String):void ▲ b id: int ▲ p_name: String ▼TtableDTO() setRr_id(String):void setR_id(String):void ▲ p_name: String getGender():String ▲ m_id: String A price: int getR name():String getR name():String getT id():String ▲ rdate: String setGender(String):void ▲ title: String PtableDTO() setT_id(String):void setR_name(String):void setR_name(String):void A price: int getBirth():String ▲ content: String getP_id():String getR_id():String getSr_id():String getLocation():String ▲ m_id: String setBirth(String):void ▲ uDate: Timestamp setP id(String):void setR_id(String):void setSr_id(String):void setLocation(String):void ▲ m_name: String getTall():int A hits: int getP name():String getR_name():String getRdate():String getPeople():int ▲ kdate: Timestamp setTall(int):void setRdate(String):void setP_name(String):void setR_name(String):void setPeople(int):void ●BoardDTO() **SmartRDTO()** getTel():String getPrice():int getStime():String getStime():String ◆BoardDTO(String, String, String, Timestamp, int) getR_ltall():int getSr_id():String setTel(String):void setPrice(int):void setStime(String):void setStime(String):void getB_id():int setR_ltall(int):void setSr_id(String):void getEmail():String getFtime():String getFtime():String getR_htall():int setB_id(int):void getP id():String setEmail(String):void setFtime(String):void setFtime(String):void setR htall(int):void getM_id():String setP_id(String):void netBtime():String and Rtime(): String a satM id/String) world a net! time() int

(4) SwingLJT, 게시판 – DTO, DAO 연결상태 **1-5) UML**

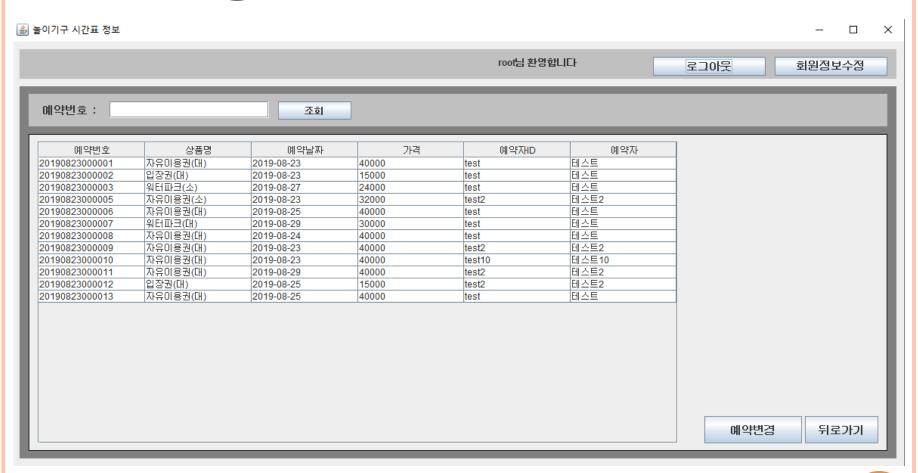


2-3] CRUD - 정병수

📤 놀이기구 시간표 정보	▲ 놀이기구 시간표 정보	_ ×	- □ X
			로그아뭇 회원정보수정
예약 날짜 : 2019-08-23	예약 정보를 확인 해주세요.		
순번 1 2 3 4	예약자명 : 관리자		
5 6 7 8	예약상품: 자유이용권(대)	4	
	이용날짜: 2019-08-23	- 1	
	가격: 40000원	- 1	
	확인 취소		추가 수정/삭제 예약하기 뒤로가기

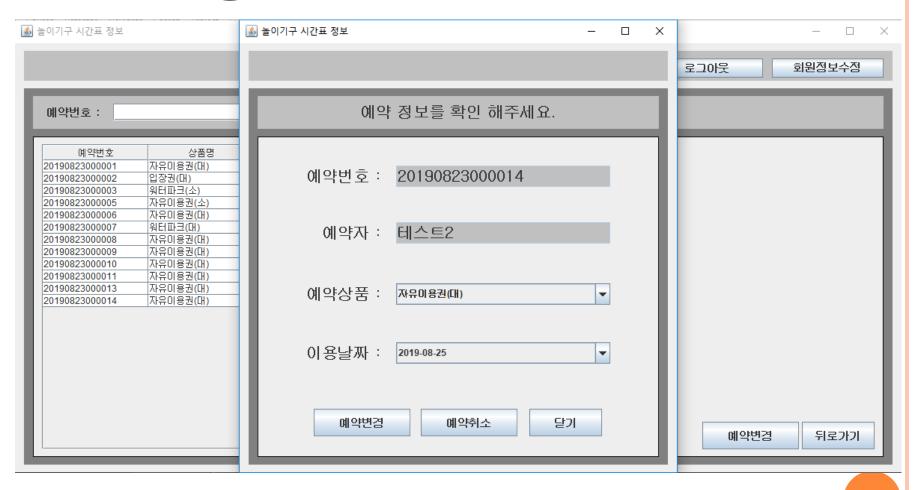
텍스트 필드에서 DTO를 통해 넘겨 받은 값으로 Insert문 실행하여 예약하는 화면

2-3) CRUD - 정병수



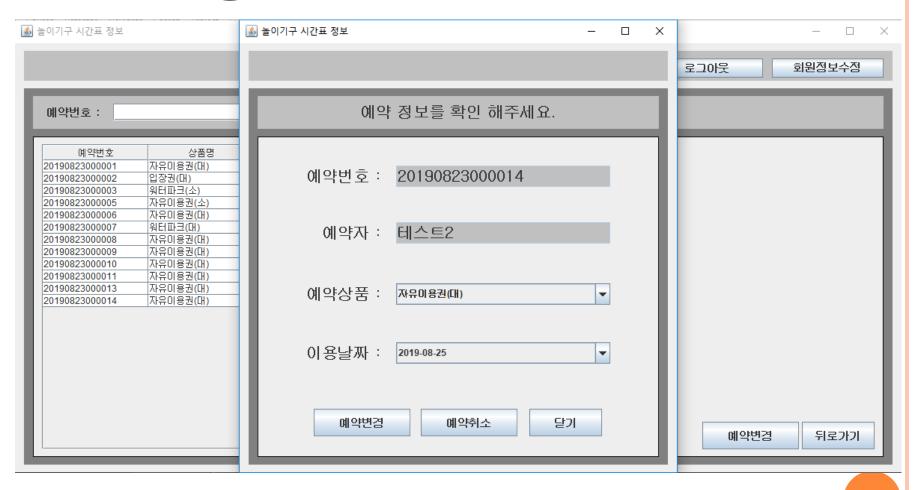
Select문 결과의 각 레코드를 DTO에 넣고 전체를 ArrayList로 받아주어 DefaultTableModel에 Row추가하여 조회결과를 보여주는 화면

2-3) CRUD - 정병수



텍스트 필드에 있는 예약정보를 받아 Update문을 실행하여 예약을 변경하는 화면

2-3) CRUD - 정병수



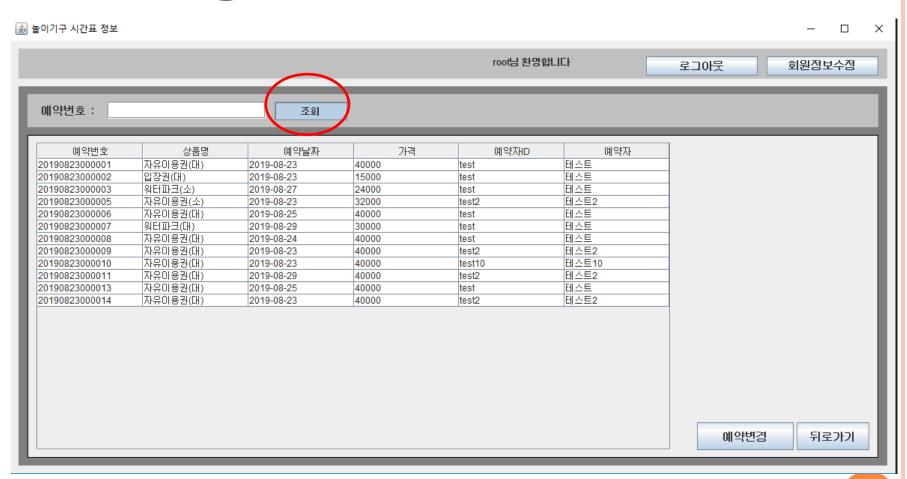
텍스트 필드에 있는 예약번호(키값)를 받아 Delete문을 실행하여 해당 예약 내역을 취소하는 화면

2-3) 예약번호 알고리즘 - 정병수

ABŞ SR_id ₹‡	ANE P_id ₹‡	ABC P_name 🏋	ABC rDate ₹‡	123 price \(\frac{1}{4}\)	ANE W_id ₹‡	ABC M_name T‡		T:
20190823000001	☑ a708	자유이용권(대)	2019-08-23	40,000	☑ test	테스트	2019-08-23 11	:48:31
20190823000002	☑ b708	입장권(대)	2019-08-23	15,000	test	테스트	2019-08-23 11	:48:37
20190823000003	☑ d708s	워터파크(소)	2019-08-27	24,000	☑ test	테스트	2019-08-23 11	:48:44
20190823000005	☑ a708s	자유이용권(소)	2019-08-23	32,000	☑ test2	테스트2	2019-08-23 11	:49:06
20190823000006	☑ a708	자유이용권(대)	2019-08-25	40,000	☑ test	테스트	2019-08-23 12	2:03:20
20190823000007	☑ d708	워터파크(대)	2019-08-29	30,000	test	테스트	2019-08-23 12	:03:24
20190823000008	☑ a708	자유이용권(대)	2019-08-24	40,000	☑ test	테스트	2019-08-23 12	:22:21
20190823000009	☑ a708	자유이용권(대)	2019-08-23	40,000	☑ test2	테스트2	2019-08-23 12	:24:10
20190823000010	☑ a708	자유이용권(대)	2019-08-23	40,000	☑ test10	테스트10	2019-08-23 12	:27:22
20190823000011	☑ a708	자유이용권(대)	2019-08-29	40,000	☑ test2	테스트2	2019-08-23 12	2:33:39
20190823000013	☑ a708	자유이용권(대)	2019-08-25	40,000	☑ test	테스트	2019-08-23 12	2:34:38
20190823000014	☑ a708	자유이용권(대)	2019-08-23	40,000	☑ test2	테스트2	2019-08-23 18	3:02:58

DateTimeFomatter로 날짜형식과 순차적으로 증가하는 6자리 format을 조합하여 예약번호 생성. 예약번호 생성 시 최근 예약번호를 불러와 당일 예약이 있으면 +1, 없으면 초기값(1)로 예약번호 생성

2-3) 팝업 변경 Reflash - 정병수



조회 버튼 클릭 시 Select문 조건을 like로 주어 텍스트 필드가 빈 값일 때모든 예약 내역 출력. 조회 완료 후 텍스트 필드(검색창) 초기화. WindowFocusListener매서드로 화면 선택 시 조회 버튼 클릭 되도록 정의

" 감사합니다. " Q&A?



