3조 프로젝트 보고서

2022 년 11 월 25 일

역할	이름
조장	임다솜
조원	노설아
조원	정재웅
조원	손일형

목차

- 1. 조원 소개
- 2. 프로젝트 Timeline
  - 2-1 프로젝트 요약 Timeline
- 3. 요구사항 충족 여부
  - 3-1 요구사항 분석
  - 3-2 요구사항 충족 여부
- 4. 개발환경
- 5. 주제 선정이유 및 목적
  - 5-1 주제 선정이유 : 구독 서비스 시장의 급격한 증가
  - 5-2 프로젝트 목적
- 6. 기획
  - 6-1 기능
  - 6-2 구상도
  - 6-3 UI 구상도
- 7. 데이터 모델링
- 8. 클래스 다이어그램
  - 8-1. 전체 클래스 다이어그램
  - 8-2. 중요 클래스 다이어그램 세부 설명
  - 8-3. 클래스 세부 다이어그램
- 9. 스토리보드
  - 9-1. 로그인
  - 9-2.회원가입
  - 9-3.회원정보수정
  - 9-4. 구독서비스보기
  - 9-5. 구독서비스보기
  - 9-6, 나의 구독 서비스
- 10. 테이블 명세서
- 11. ER 다이어그램
- 12. 후기

1. 조원 소개

조장 임다솜

[로그인 / 회원가입 / 회원정보 수정 / 나의 구독 추가 / 메인메뉴] 페이지 코드구현

프로젝트 총괄, DB설계, CSS 구현, FXML 레이아웃 수정, 보고서 작성

조원 노설아

[나의 구독서비스 보기] 페이지 코드 구현

DB 테이블 자료 조사, PPT 제작, 프로그램 시연, 보고서 작성

조원 정재웅

[구독서비스 추가] 페이지 코드 구현

프로그램 시연, 보고서 작성, DB 테이블 자료 조사, 보고서 작성,

CSS 구현

조원 손일형

[구독서비스 보기] 페이지 코드 구현

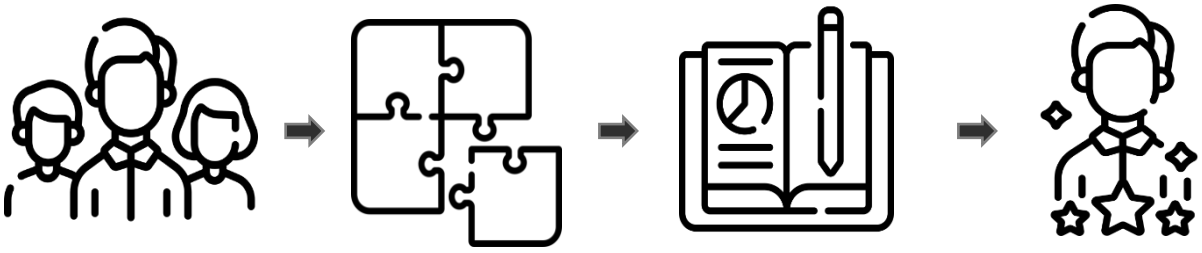
UI 스케치 제작, SQL VIEW 작성, PPT 탬플릿 제작 및 PPT 내용 작성,

보고서 탬플릿 제작 및 내용 작성

영상 및 자막 편집 작업

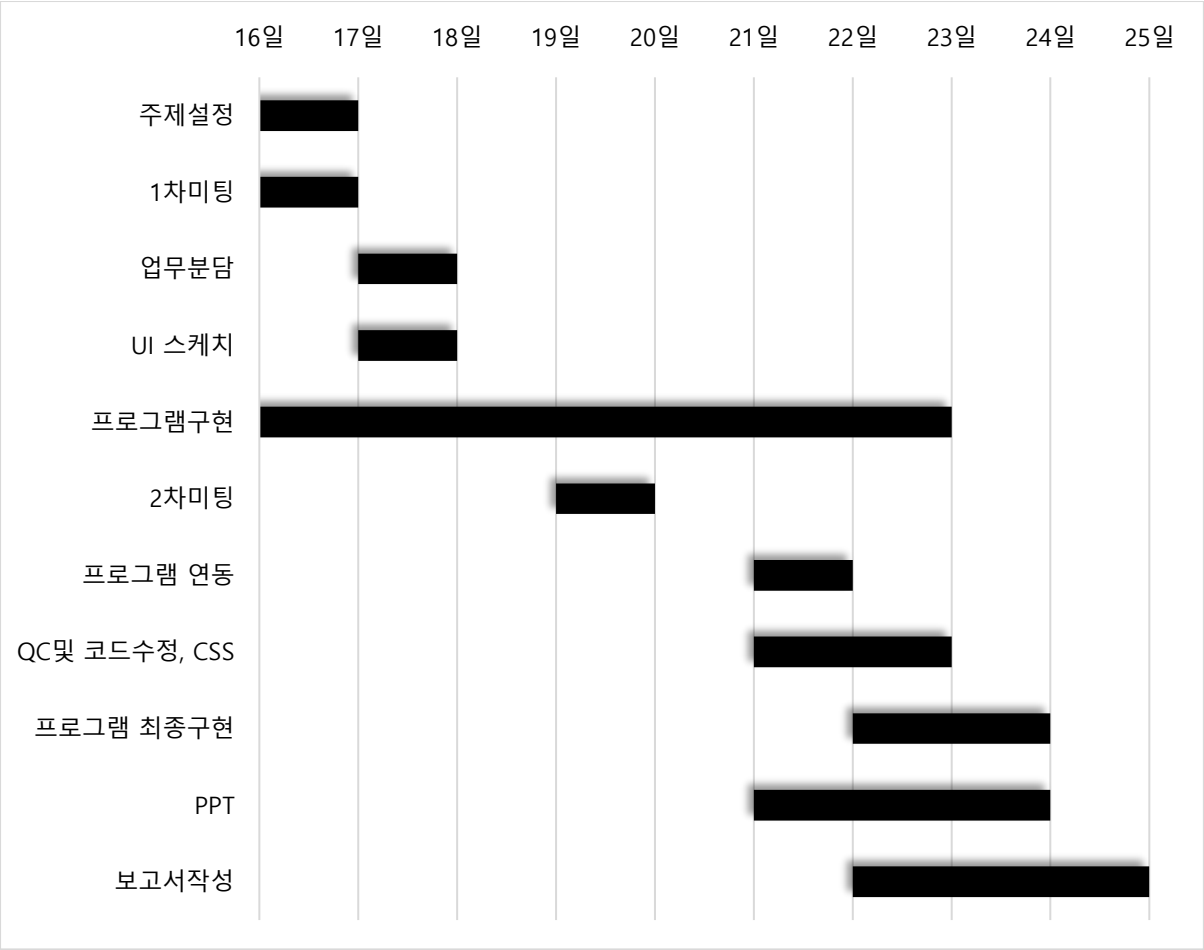
2. 프로젝트 Timeline

2-1 프로젝트 요약 Timeline



개발 → **테스트 및 오류 수정** → **서류작업** → **디자인 및 발표 준비**

2-2. 프로젝트 상세 Timeline



3. 요구사항 충족 여부

3-1 요구사항 분석

요구사항 분류	요구사항 명칭	요구사항 세부내용	비고
기능 요구사항	구독 서비스 보기	-db에 등록되어 있는 구독서비스 조회 기능 -리스트 형식으로 출력 -카테고리별 조회 기능 -카테고리 목록 : 전체, 음악, 영상, 게임, 서적 -리스트 목록을 클릭하여 상세 정보 조회 기능 -로그인한 유저의 구독 여부 확인 할수 있는 기능 -리스트에 보여지는 형식 : 카테고리, 구독 서비스명, 아이콘 , 구독여부	
	구독 서비스 추가	-로그인한 유저의 새로운 구독 추가 -서비스명 선택 가능 -카테고리 choice box로 선택 -요금제 입력 가능 -구독인원 입력 가능 -상세정보 입력 기능 -작성된 형식을 db에 저장	
	회원가입	-새로운 회원을 등록한다. -id, 비밀번호, 이름, 이메일 등록 가능 -비밀번호 재확인 기능이 필요함 -email 중복확인 기능 필요	

요구사항 분류	요구사항 명칭	요구사항 세부내용	비고
기능 요구사항	로그인	-등록된 회원정보를 이용해 서비스에 접속 -id와 password를 이용하여 인증 -입력한 id가 등록되어 있지 않다면 유저에게 알림 -입력한 pw가 저장된 pw 와 다르다면 유저에게 알림 -입력한 id와 pw가 등록되어 있는 정보와 같다면 서비스 홈 화면으로 전환 -인증된 유저의 정보를 지속적으로 확인 가능	
프로젝트 제약사항	JAVA 기술요구사항	-java 8 -DAO pattern -factory pattern -JDBC -ResultSet 사용 -PreparedStatement 사용 -JAVA FX -JAVA FX Scene Builder -FXML 레이아웃 사용	
	코드 작성요구사항	-패키지 2개 이상 사용 -모든 java 소스에 주석 작성 -코드작성자 표시(공동작성자가능) -페어 프로그래밍 -참조 소스 원본 표시	
	Database 요구 사항	-Oracle DB (11G or 18CEx or Cloud) -View 사용 -새로운 유저 생성	

3-2 요구사항 충족 여부

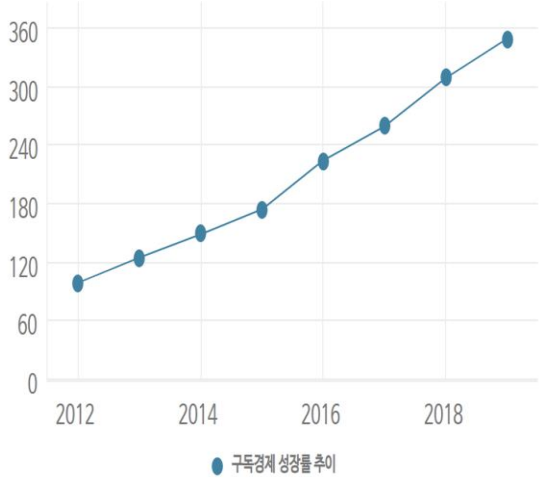
UI 요구사항	필수여부	충족여부
로그인, 회원가입 구현	필수	○
추가 기능 페이지 구현	필수	○
JAVA 기술 요구사항		
JAVA 8	필수	○
DAO pattern	필수	○
factory pattern (interface 사용)	권장	○
JDBC	필수	○
Resultset 사용	필수	○
PreparedStatement 사용	필수	○
JAVA FX	필수	○
JAVA FX Scene Builder	필수	○
FMXL 레이아웃 사용	필수	○
코드 작성시 요구사항		
패키지 2 개 이상 사용	권장	○
코드 작성자 표시(공동작성자가능)	권장	○
페어 프로그래밍	필수	○
DATABASE 요구사항	필수	○
ORACLE DB(11G OR 18CEx 0R Cloud)	필수	○
View 사용	필수	○
새로운 유저 생성	필수	○
새로운 테이블 생성	필수	○

4. 개발환경

- **운영체제** : **Windows 10 Pro**
- **DBMS** : **Oracle Database 18c Express Edition**
- **개발 및 설계 도구** : **Eclipse, Oracle SQL developer20.4, SceneBuiler8.5.0**
- **개발 언어** : **Java 1.8, JavaFX**
- **외부 라이브러리** : **OJDBC7, JBCRYPT0.4.1**

5. 주제 선정이유 및 목적

5-1 주제 선정이유 : 구독 서비스 시장의 급격한 증가



*자료: Zuora(2019), 「Zuora Subscription Economy Index」

현대사회는 바야흐로 구독경제(Subscription Economy)의 시대라고 해도 과언이 아닐 것이다. 우리는 아침에 출근을 하며 'Melon'을 통해 음악을 듣고, 퇴근길엔 'Netflix'로 예능 프로그램이나 챙겨보지 못한 드라마들을 감상한다. 그 외에도 수많은 구독서비스들은 이미 우리에게 친숙한 존재다.

기존에 단순히 음악, 영상 등 미디어컨텐츠에 국한하던 구독 서비스는 나아가서 식품과 가전제품, 차량까지 다양한 영역에서 구독경제 모델이 등장하고 있다. 코로나 팬데믹으로 인한 비대면 수요의 증가, 세계적 경기침체에 따른 물건을 직접 구매하기보단 필요할 때만 구독해서 사용하는 등 시장의 성장 잠재력은 충분하고 실제로 많은 기업들이 구독 서비스에 발을 넓히고 있다.

이렇게 일상 속에 자리잡아 있는 다양한 구독 서비스가 가지고 있는 특징 중 하나는 월별, 혹은 년 단위 별로 나도 모르는 사이 정기적으로 결제가 된다는 것이다. 과연 구독 서비스 이용자 중 구독하고 있는 서비스의 종류는 몇 개나 되는지 아는 사람은 있어도 그것들이 각각 얼마씩 결제되고 한달에 그로 인해

지출하는 비용이 얼마인지 아는 사람은 얼마나 될까? 각각 다른 가격, 다른 날짜에 결제되는 서비스들을 전부 파악하는 것은 쉽지 않다. 더군다나 요즘엔 구독서비스의 공유도 활성화되어 원래 요금이 아닌 1/n 으로 요금을 나눠내는 것도 흔하기에 더욱 모를 것이다.

5-2 프로젝트 목적

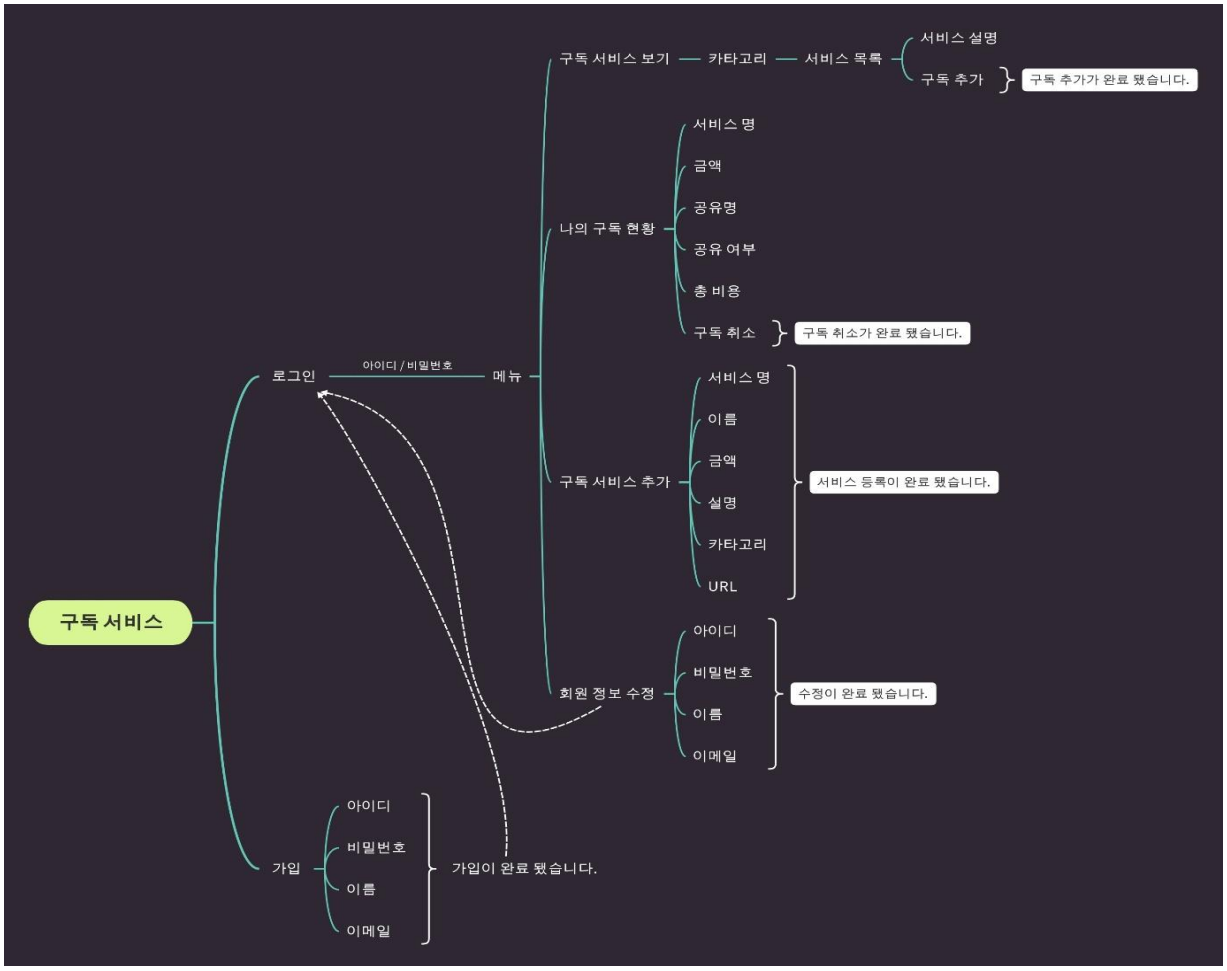
상기에 적은 것 같이 많은 사람들이 적지 않은 금액을 구독서비스에 사용하면서 정확하게 얼마를 쓰는지 아는 사람은 많지 않다. 이런 생각으로부터 우리는 "내가 구독하는 서비스들을 한눈에 보고 금액까지 알 수 있으면 어떨까?" 라는 생각을 본 프로젝트의 주 목적으로 삼게 되었다. 추가적인 기능으로 국내에서 이용하는 가능한 다양한 구독서비스 들을 카테고리 별로 사용자에게 보여주어 나에게 맞는 구독 서비스를 찾아 추가할 수 있는 기능을 구현하기로 하였다.

6. 기획

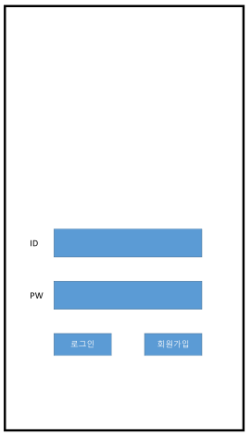
6-1 기능

필수기능	추가기능
로그인	비밀번호 암호화
회원가입	구독서비스 카테고리 분류
구독서비스 정보보기	구독서비스 추가
나의 구독서비스 보기	회원정보 수정
나의 구독서비스 추가	CSS 사용

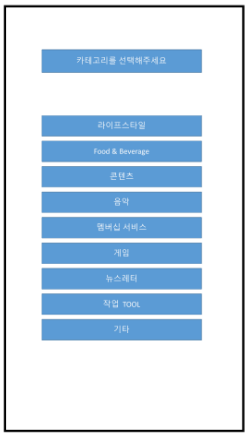
6-2 구상도



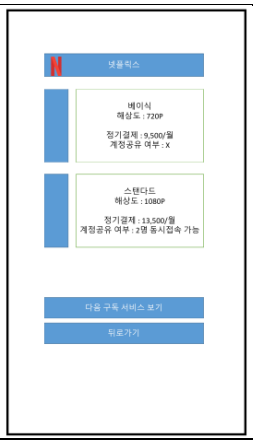
6-3 UI 구상도



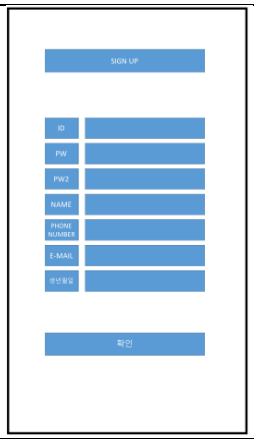
로그인



메뉴

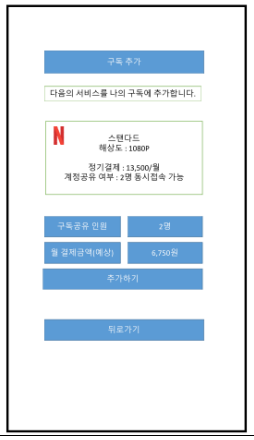


나의 구독 현황



구독 서비스 추가

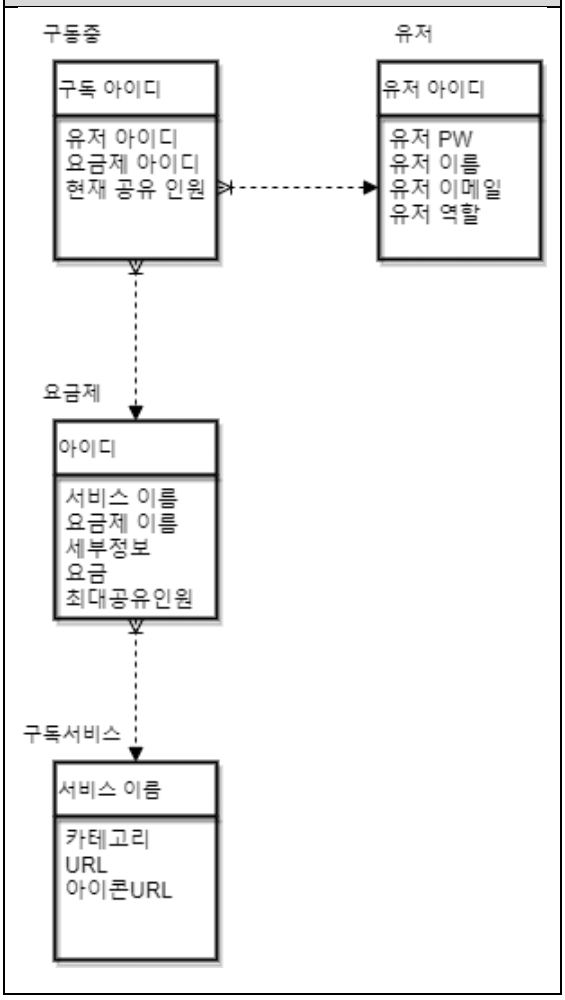
구독 상세정보



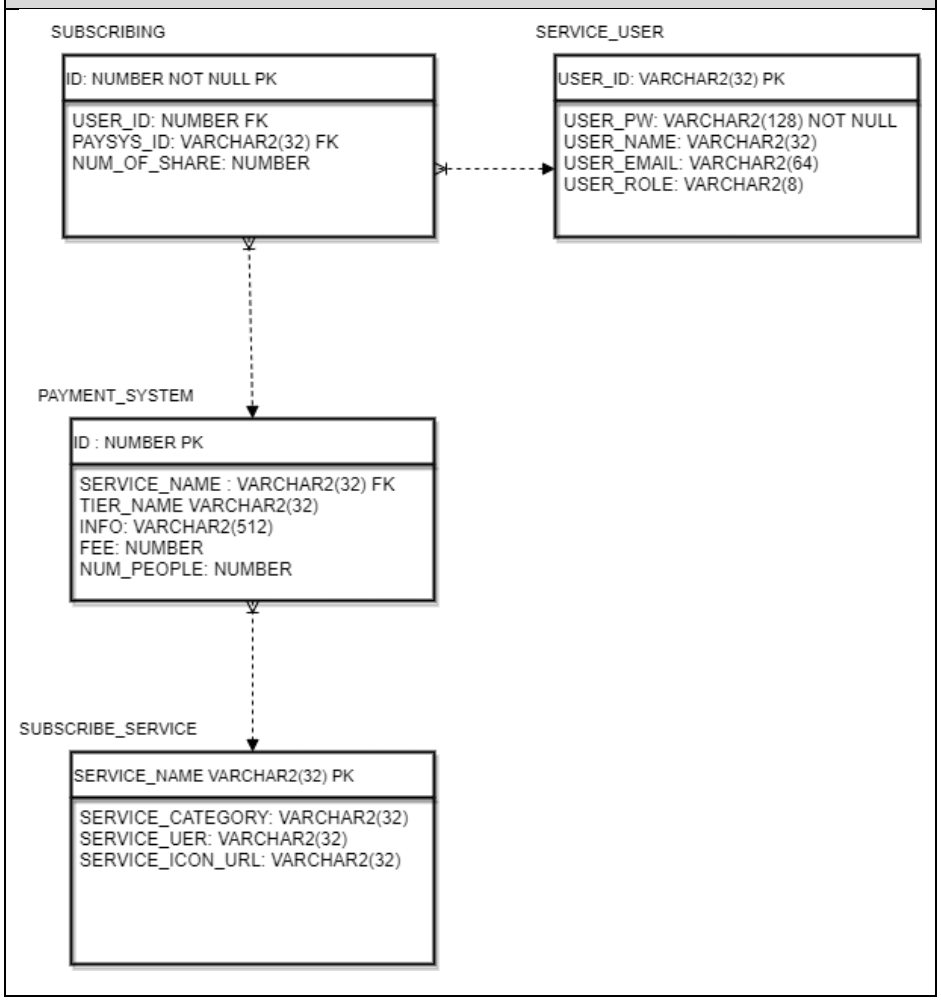
구독 추가

7. 데이터 모델링

논리적 모델링

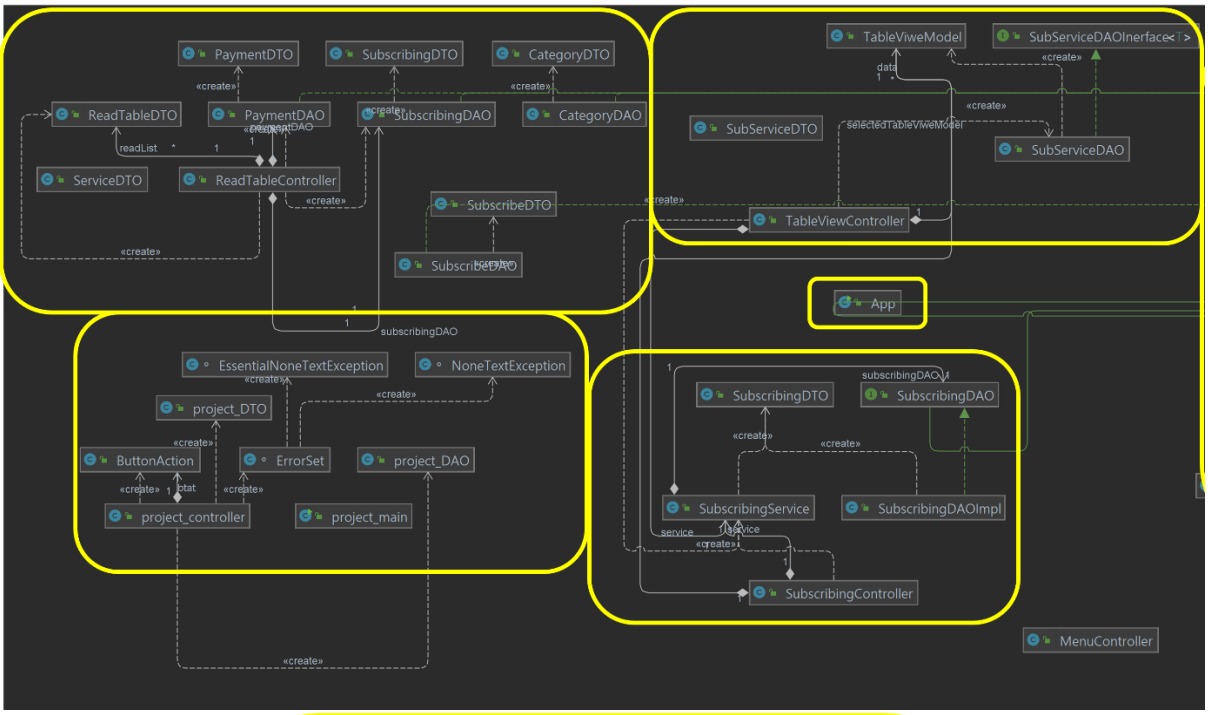


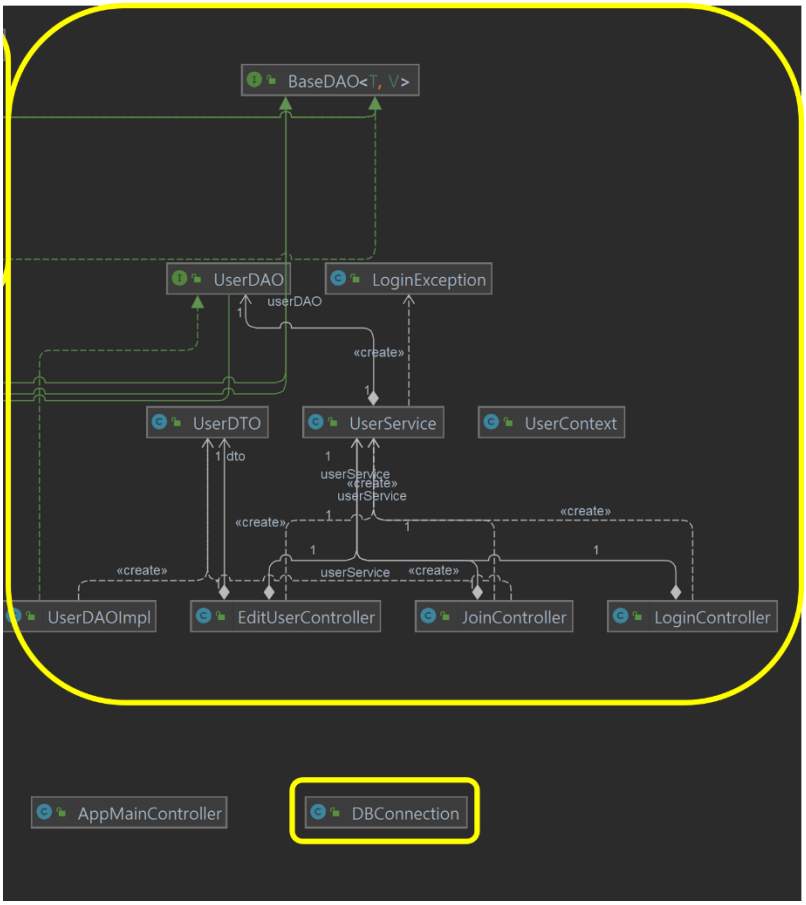
물리적 모델링



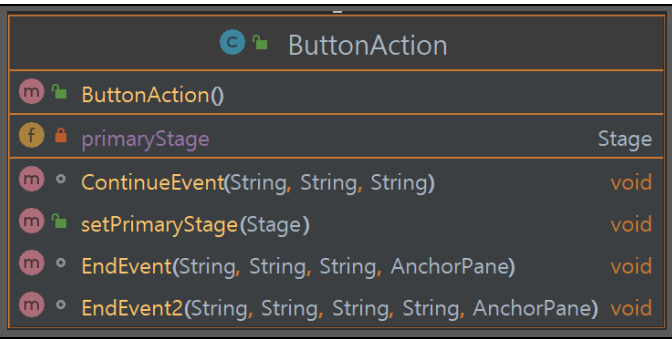
8. 클래스 다이어그램

8-1. 전체 클래스 다이어그램





8-2. 중요 클래스 다이어그램 세부 설명

ButtinAction			
주요 메소드	**주요 코드**	**상세설명**	**비고**

ContinueEvent	Button btnOk = (Button) parent.lookup("#btnOk");
	//버튼 위치 ID 로 찾아 가기
	btnOk.setOnAction(event -> {
	// 버튼 이벤트 실행
	dialog.close();
	//버튼을 누를시 팝업창 닫기
	});
	
		버튼 이벤트 실행시 팝업창만 닫힘.	

EndEvent	btnOk.setOnAction(event -> { // 버튼 이벤트 실행
	dialog.close();//버튼을 누를시 팝업창 닫기
	Pane appMain = (Pane) addService.getParent();//팝업창의 위치 찾기
	appMain.getChildren().clear();// 팝업창이 닫아질때 이전 창 까지 같이 닫이기
	try {
	Parent newPane = FXMLLoader.load(getClass().getResource("../menu/main.fxml"));
	appMain.getChildren().add(newPane);
	} catch (Exception e3) {
	// TODO: handle exception
	}
	}
		버튼 이벤트 실행시 팝업창이 닫히고 지정된 메인 화면으로 화면이 넘어감.	

EndEvent2	btnOk.setOnAction(event -> { // 버튼 이벤트 실행
	dialog.close();//버튼을 누를시 팝업창 닫기
	Pane appMain = (Pane) addService.getParent();//팝업창의 위치 찾기
	appMain.getChildren().clear();// 팝업창이 닫아질때 이전 창 까지 같이 닫이기
	try {
	Parent newPane = FXMLLoader.load(getClass().getResource("../menu/main.fxml"));
	appMain.getChildren().add(newPane);
	} catch (Exception e3) {
	// TODO: handle exception
	}
	}
		버튼 이벤트 실행시 팝업창이 닫히고 지정된 메인 화면으로 화면이 넘어감.	

project_DAO			
	project_DAO project_DAO() conn Connection categoryStringList() List<String> findByTier(String, String) boolean Payment_Insert(project_DTO) void findByService(String) boolean Service_Insert(project_DTO) void		
주요 메소드	**주요 코드**	**상세설명**	**비고**

categortStringList	String query = "select distinct service_category as category from subscribe_service";//select SQL 문
	List<String> categoryList = new ArrayList<>();//ArrayList 생성
	try {
	PreparedStatement pstmt = conn.prepareStatement(query);
	ResultSet rs = pstmt.executeQuery();
	while(rs.next())// 값이 없을때 까지 반복
	categoryList.add(rs.getString("CATEGORY"));//배열에 저장
	}
		SQL SELECT 문을 활용하여 카테고리의 값을 찾고 distinct 로 중복 값을 제거 이후 이 값을 ResultSet 으로 저장하여 카테고리 목록배열 생성	

findByTier	String query = "select count(*) as count from PAYMENT_SYSTEM where service_name = ? and tier_name = ?";//select SQL 문
	try {
	PreparedStatement pstmt = conn.prepareStatement(query);//service_name, tier_name select 부분
	pstmt.setString(1, serviceName);//service_name
	pstmt.setString(2, tier);//tier_name
	ResultSet rs = pstmt.executeQuery();
	rs.next();
	if(rs.getInt("count") > 0)// 중복된 값이 있을 경우 true
	return true;
		SQL SELECT 문을 활용하여 PAYMENT_SYSTEM 에 servicename 및 tiername 이 있을 경우 true 를 반환 아무것도 없을 경우 false	

Paument_Insert	String PAYMENT_sql = "insert into PAYMENT_SYSTEM (service_name, tier_name, info, fee, num_people) VALUES (?,?,?,?,?) ";//DB PAYMENT_SYSTEM INSERT 부분
	try {
	PreparedStatement PAYMENT_preparedStatement = conn.prepareStatement(PAYMENT_sql);//PAYMENT_SYSTEM INSERT 부분
	PAYMENT_preparedStatement.setString(1, newAccount.getPay_service_name());//service_name 에 들어갈 DTO getPay_service_name()
	PAYMENT_preparedStatement.setString(2, newAccount.getPay_tier_name());//tier_name 에 들어갈 DTOgetPay_tier_name()
	PAYMENT_preparedStatement.setString(3, newAccount.getPay_info());//info 에 들어갈 DTOgetPay_info()
	PAYMENT_preparedStatement.setInt(4, newAccount.getPay_fee());//fee 에 들어갈 DTOgetPay_fee()
	PAYMENT_preparedStatement.setInt(5, newAccount.getPay_num_people());//num_people 에 들어갈 DTOgetPay_num_people()
	PAYMENT_preparedStatement.executeUpdate();//Foreign Key PAYMENT_SYSTEM 업데이트
		PreparedStatement 를 활용하여 데이터 입력 값을 받아 SQL 문 insert findByService 의 중복 여부에 따라 작동	

findByService	String query = "select count(*) as count from SUBSCRIBE_SERVICE where service_name = ?";//select SQL 문
	try {
	PreparedStatement pstmt = conn.prepareStatement(query);//service_name
	pstmt.setString(1, serviceName);//service_name
	ResultSet rs = pstmt.executeQuery();
	rs.next();
	
	if(rs.getInt("count") > 0)// 중복된 값이 있을 경우 true
	{
	return true;
	}
		SQL SELECT 문을 활용하여 SUBSCRIBE_SERVICE 에 servicename 이 있을 경우 true 를 반환
		아무것도 없을 경우 false	

Service_Insert	String SERVICE_sql = "insert into SUBSCRIBE_SERVICE (service_name, service_category, service_URL, service_icon_URL) VALUES (?, ?, ?,?)";//DB SUBSCRIBE_SERVICE INSERT 부분
	String PAYMENT_sql = "insert into PAYMENT_SYSTEM		

	(service_name, tier_name, info, fee, num_people) VALUES (?,?,?,?,?) ";//DB PAYMENT_SYSTEM INSERT 부분
	
	try {
	PreparedStatement SERVICE_preparedStatement = conn.prepareStatement(SERVICE_sql);//SUBSCRIBE_SERVICE INSERT 부분
	SERVICE_preparedStatement.setString(1, newAccount.getSv_service_name());//service_name 에 들어갈 DTO getSv_service_name()
	SERVICE_preparedStatement.setString(2, newAccount.getSv_category());//service_category 에 들어갈 DTO getSv_category()
	SERVICE_preparedStatement.setString(3, newAccount.getPay_URL());//service_URL 에 들어갈 DTO getPay_URL()
	SERVICE_preparedStatement.setString(4, newAccount.getPay_icon_URL());//service_icon_URL 에 들어갈 DTO getPay_icon_URL()
	SERVICE_preparedStatement.executeUpdate();//Primary Key SUBSCRIBE_SERVICE 업데이트
	
	PreparedStatement PAYMENT_preparedStatement = conn.prepareStatement(PAYMENT_sql);//PAYMENT_SYSTEM INSERT 부분
	PAYMENT_preparedStatement.setString(1, newAccount.getPay_service_name());//service_name 에 들어갈 DTO getPay_service_name()
	PAYMENT_preparedStatement.setString(2, newAccount.getPay_tier_name());//tier_name 에 들어갈 DTOgetPay_tier_name()
	PAYMENT_preparedStatement.setString(3, newAccount.getPay_info());//info 에 들어갈 DTOgetPay_info()
	PAYMENT_preparedStatement.setInt(4, newAccount.getPay_fee());//fee 에 들어갈 DTOgetPay_fee()
	PAYMENT_preparedStatement.setInt(5, newAccount.getPay_num_people());//num_people 에 들어갈 DTOgetPay_num_people()
	PAYMENT_preparedStatement.executeUpdate();//Foreign Key PAYMENT_SYSTEM 업데이트
		preparedStatement 를 활용하여 데이터 입력값을 받아 SQL 문 insert
		findByService 의 중복 여부에 따라 작동	

project_controller /ErrorSet	project_controller project_controller() Category_ComboBox ComboBox sv_service_category String Service_Name TextField Tier TextField sv_service_url String sv_service_name String Num_People TextField btat ButtonAction Icon_URL Button sv_service_icon_URL String pay_service_name String pay_service_info String pay_tier_name String primaryStage Stage Fee TextField Service_Info TextArea addService AnchorPane pay_num_people int pay_fee int Service_URL TextField	handleService_CommitAction(ActionEvent) void handleIcon_URL(ActionEvent) void setPrimaryStage(Stage) void handleService_RollbackAction(ActionEvent) void initialize(URL, ResourceBundle) void ErrorSet ErrorSet() errorChk(int, int, int, int, int, int, int) void		
주요 메소드	**주요 코드**	**상세설명**	**비고**

initialize	project_DAO dao = new project_DAO();
	ObservableList<String> fxComboNameList = FXCollections.observableArrayList(dao.categoryStringList());// String 배열에 값 전달
	Category_ComboBox.setItems(fxComboNameList);// Combobox 에 리스트 전달
		DAO 에서 배열로 저장된 값을 전달받아 ComboBox 리스트로 지정	

handleIcon
_URL
	FileChooser fileChooser = new FileChooser();//FileChooser 생성
	fileChooser.getExtensionFilters().addAll(
	new ExtensionFilter("Image Files", "*.png", "*.jpg", "*.gif"));// 파일 열기 형식 지정
	File selectedFile =
	fileChooser.showOpenDialog((Stage)addService.getScene().getWindow());//Dialog
	if (selectedFile != null) {//파일을 선택 했을 경우
	sv_service_icon_URL=selectedFile.getPath();//sv_service_icon_URL 파일 주소값 저장
	System.out.println(selectedFile.getPath());//확인 프린트
	btat.ContinueEvent("Add_Service_Popup.fxml","선택 됐습니다.","확인 ");//팝업창으로 확인
	}
	else if(selectedFile == null)//파일을 선택 하지 않을 경우
	sv_service_icon_URL= null;//null 값 저장
		fileChooser 을 사용하여 파일 불러오기 기능, 사용 버튼 이벤트와 연동하여 버튼을 누를 시 파일 불러오기 활성화
		이미지 파일만 가능하게 ExtensionFilter 에서 파일형식 지정, 파일을 선택 시 변수에 경로 저장, 선택하지 않을 시에는 null 값 저장	

handleService
_CommitAction
	/*필수 저장 이외의 장소에서는 값을 기입 하지 않았을 경우 null 값 및 0 이 저장 되게 설정, 설정 하지 않으면 아스키코드 값으로 저장 됨*/
	if (lenNP == 0)//공유 가능 인원 기입 하지 않을 경우 숫자 1 저장
	pay_num_people =1;
	else
	pay_num_people = Integer.parseInt(Num_People.getText());// 공유 인원 저장
	if (lenSU == 0) //URL 기입 하지 않을 경우 null 저장
	sv_service_url = null;
	else
	sv_service_url = Service_URL.getText();//URL 저장
	if (lenSI == 0) //상세정보 기입 하지 않을 경우 null 저장
	pay_service_info = null;
	else
	pay_service_info = Service_Info.getText();//상세 설명 저장
		필수 저장 이외의 장소에서는 값을 기입하지 않았을 경우 null 값 및 0 이 저장되게 설정, 설정하지 않으면 아스키코드 값으로 저장	

ErrorSet	void errorChk (int lenSN, int lenSI, int lenSU, int lenSC, int lenFee, int lenNP, int lenTN) throws NoneTextException, EssentialNoneTextException
	{
	//필수 입력값 누락시 프린트 서비스명, 카타고리, 가격, 공유 가능 인원
	if ((lenSI > 0 || lenSU > 0 || lenNP > 0) && (lenSN == 0 || lenSC == 0 || lenFee == 0 || lenTN == 0))
	throw new EssentialNoneTextException("필수 입력값이 누락 됐습니다.");
	//전체적으로 입려값이 없을때
	else if(lenSN == 0 || lenSC == 0 || lenFee == 0 || lenTN == 0 ||lenSN == 0 || lenSC == 0 || lenFee == 0 || lenTN == 0)
	throw new NoneTextException("입력값이 없습니다.");
	}
		필수 입력 값 누락시 프린트서비스명, 카테고리, 가격, 공유가능인원 전체적으로 입력값 없을때 사용자 예외처리 지정	

TableViewController	subtableview TableView<TableViweModel> share TableColumn<TableViweModel, Integer> service_name TableColumn<TableViweModel, String> fee_name TableColumn<TableViweModel, String> fee TableColumn<TableViweModel, Integer>		
주요 변수	**주요 코드**	**상세설명**	**비고**
subtableview	@FXML private TableView<TableViweModel> subtableview;	@FXML 어노테이션을 사용해 테이블뷰와 각 열을 연결시킴	
service_name	@FXML private TableColumn<TableViweModel, String> service_name;		테이블뷰에 종속된 열
fee_name	@FXML private TableColumn<TableViweModel, String> fee_name;		
info	@FXML private TableColumn<TableViweModel, String> info;		
fee	@FXML private TableColumn<TableViweModel, Integer> fee;		
share	@FXML private TableColumn<TableViweModel, Integer> share;		

TableViewController	setCellTable() void BTNall(ActionEvent) void BTNmusic(ActionEvent) void BTNmovie(ActionEvent) void BTNgame(ActionEvent) void BTNcult(ActionEvent) void		
주요 메소드	**주요 코드**	**상세설명**	**비고**

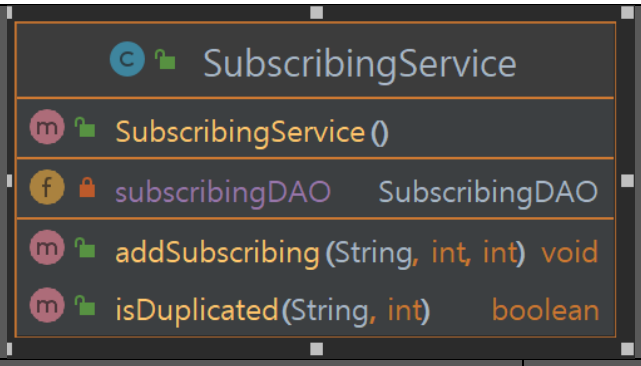
setCellTable()	private void setCellTable() {
	service_name.setCellValueFactory(new PropertyValueFactory<>("serviceName"));
	fee_name.setCellValueFactory(new PropertyValueFactory<>("fee_name"));
	info.setCellValueFactory(new PropertyValueFactory<>("info"));
	fee.setCellValueFactory(new PropertyValueFactory<>("fee"));
	share.setCellValueFactory(new PropertyValueFactory<>("share"));
	}
		테이블뷰 속 cell 에 들어갈 데이터 형식을 TableViewModel 의 Property 로 세팅함	

BTNall 外 4 개	public void BTNall(ActionEvent e) {
	SubServiceDAO DAO = new SubServiceDAO();
	List<TableViweModel> serviceVOs = DAO.dataList();
	
	observableList = FXCollections.observableArrayList(serviceVOs);
	setCellTable();
	subtableview.setItems(observableList);
	}
		각 버튼에 맞는 액션이벤트를 실행 시키는 메소드
		DB 에서 가져온 colunm 을 테이블뷰에 불러옴	

SubServiceDAO	dataList() List<TableViweModel> selectCategory(String) List<TableViweModel>		
주요 메소드	주요 코드	상세설명	비고

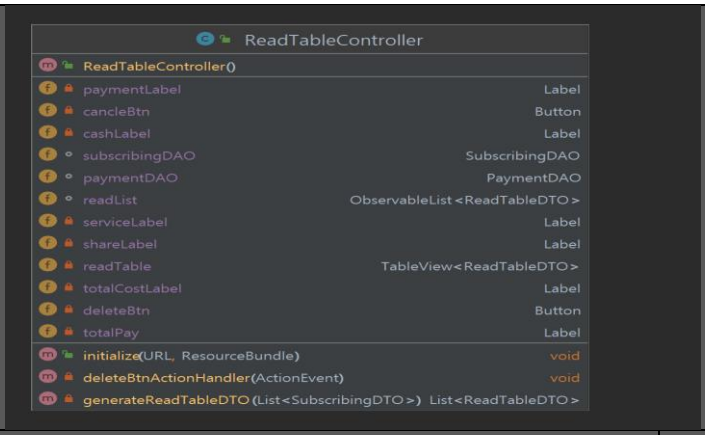
dataList()	public List<TableViweModel> dataList() {
	String sql = "select * from v_catelist";
	List<TableViweModel> serviceVOs = new ArrayList<>();
	try {
	PreparedStatement preparedStatement = conn.prepareStatement(sql);
	ResultSet resultSet = preparedStatement.executeQuery();
	while (resultSet.next()) {
	serviceVOs.add(new TableViweModel(resultSet.getString(1), resultSet.getString(2), resultSet.getString(3),
	resultSet.getInt(4), resultSet.getInt(5),resultSet.getInt("id")));
	}
		전체 카테고리를 출력하는 메소드	

selsectCategory()	public List<TableViweModel> selectCategory(String insertcategory) {
	String sql = "select * from v_catelist WHERE SERVICE_CATEGORY = ?";
	List<TableViweModel> serviceVOs = new ArrayList<>();
	try {
	PreparedStatement preparedStatement = conn.prepareStatement(sql);
	preparedStatement.setString(1, insertcategory);
	ResultSet resultSet = preparedStatement.executeQuery();
	while (resultSet.next()) {
	serviceVOs.add(new TableViweModel(resultSet.getString(1), resultSet.getString(2), resultSet.getString(3),
	resultSet.getInt(4), resultSet.getInt(5),resultSet.getInt("id")));
	}
		각 카테고리별로 출력해주는 메소드	기존 코드의 경우 카테고리별로 4 개의 비슷한 코드가 존재했으나, 코드리뷰를 통해 리팩토링 제안후 ? 파라메터 사용으로 1 개의 코드문으로 줄임

SubscribingService			
주요 메소드	**주요 코드**	**상세설명**	**비고**

addSubscribing	public void addSubscribing(String user_id, int payment_id, int numOfShare) {
	SubscribingDTO subscribingDTO = new SubscribingDTO(user_id, payment_id, numOfShare);
	subscribingDAO.add(subscribingDTO);
	}
		DB 에 새로운 구독을 추가한다.	

isDuplicated	public boolean isDuplicated(String userId,int paySysId) {
	if(subscribingDAO.findByUserIdAndPaymentId(userId, paySysId) == null) return false;
	
	return true;
	}
		한 회원은 같은 요금제를 또 구독할 수 없기때문에 중복체크 하기 위한 메서드	

ReadTableController			
주요 메소드	**주요 코드**	**상세설명**	**비고**

initialize	List<SubscribingDTO> subscribingDTOs = subscribingDAO.findAllByUserId(UserContext.getInstance().getUserId());
	
	List<ReadTableDTO> readTableDTOs = generateReadTableDTO(subscribingDTOs);
	
	readTable.setFixedCellSize(-1); // readtable(filename)은 파일에서 열 방향 데이터를 읽어 테이블을 만든다
	readList = FXCollections.observableArrayList(readTableDTOs);
	
	TableColumn<ReadTableDTO,String> serviceColumn = new TableColumn<>("서비스명");
	serviceColumn.setCellValueFactory(new PropertyValueFactory<ReadTableDTO, String>("service"));
	
	TableColumn<ReadTableDTO, String> serviceNameColumn = new TableColumn<>("요금제명");
	serviceNameColumn.setCellValueFactory(new PropertyValueFactory<ReadTableDTO, String>("serviceName"));
	
	TableColumn<ReadTableDTO,Integer> cashColumn = new TableColumn<>("내 요금");
	cashColumn.setCellValueFactory(new PropertyValueFactory<ReadTableDTO, Integer>("cash"));
	
	TableColumn<ReadTableDTO, String> informationColumn = new TableColumn<>("상세정보");
	informationColumn.setCellValueFactory(new
		테이블 컬럼을 자바코드로 생성	

	PropertyValueFactory<ReadTableDTO, String>("information")); TableColumn<ReadTableDTO, Integer> shareColumn = new TableColumn<>("공유 인원"); shareColumn.setCellValueFactory(new PropertyValueFactory<ReadTableDTO, Integer>("share"));		

deleteBtnActionHandler	private void deleteBtnActionHandler(ActionEvent e) {
	
	ReadTableDTO selectedDTO = readTable.getSelectionModel().getSelectedItem();
	if(selectedDTO == null) {
	return;
	}
	System.out.println(selectedDTO.toString());
	SubscribingDAO subscribingDAO = new SubscribingDAO();
	
	subscribingDAO.deleteById(selectedDTO.getSubscringId()); /* ID 의 값을 받아서 삭제하는 문장을 넣음
	Drop 폴더 자체를 삭제한다
	Delete 는 폴더의 내용을 삭제한다 */
	
	List<SubscribingDTO> subscribingDTOs = subscribingDAO.findAllByUserId(UserContext.getInstance().getUserId());
	
	List<ReadTableDTO> readTableDTOs = generateReadTableDTO(subscribingDTOs);
	
	readList = FXCollections.observableArrayList(readTableDTOs);
	readTable.setItems(readList);
	totalPay.setText(""+SubscribingDAOImpl.getInstance().getUserTotalPay(UserContext.getInstance().getUserId()));
	}
		테이블 뷰에서 선택된 행의 데이터를 가져와서 테이블의 내용을 삭제하였습니다.	

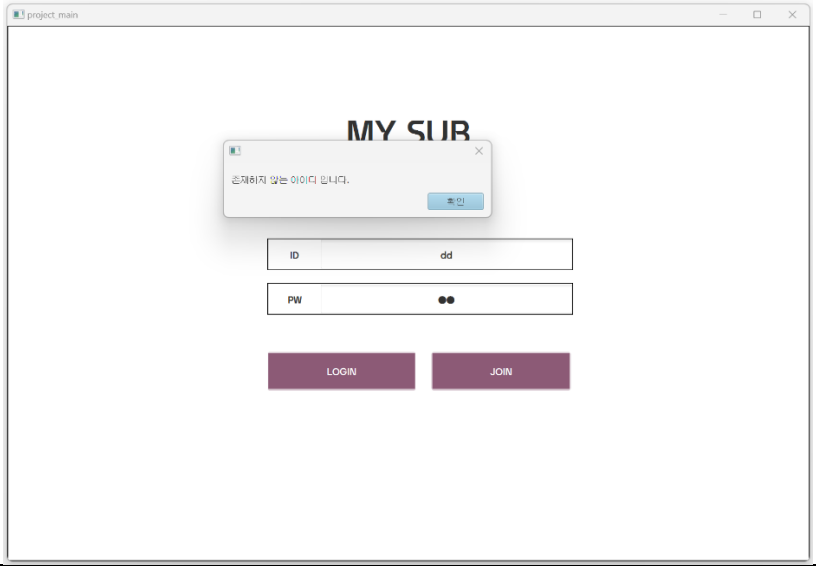
generateReadTableDTO	privateList<ReadTableDTO>
	generateReadTableDTO(List<SubscribingDTO> subscribingDTOs){
	List<PaymentDTO> paymentDTOs = new ArrayList<>();
	
	for (int i = 0; i < subscribingDTOs.size(); i++) {
	// i=0부터 시작하고 dto의 사이즈만큼만 i는 1씩 증가한다.
	PaymentDTOpaymentDTO=
	paymentDAO.findById(subscribingDTOs.get(i).getPaysys_id());
	paymentDTOs.add(paymentDTO);
	}
	
	List<ReadTableDTO> readTableDTOs = new ArrayList<>();
	for (int i = 0; i < subscribingDTOs.size(); i++) {
	ReadTableDTO
	readTableDTO=new
	ReadTableDTO(paymentDTOs.get(i).getService_name(),
	paymentDTOs.get(i).getTire_name(),paymentDTOs.get(i).getFee(),
	paymentDTOs.get(i).getInfo(),
	subscribingDTOs.get(i).getNum_of_share(),
	subscribingDTOs.get(i).getNumber());
	
	readTableDTOs.add(readTableDTO);
	
	}
	
	return readTableDTOs;
	}
		테이블뷰와 컬럼을 연결시켰는데 Join을 사용하지않고 findbyId로 데이터를 가져와서 자바코드로 대체하였습니다	

8-3. 클래스 세부 다이어그램

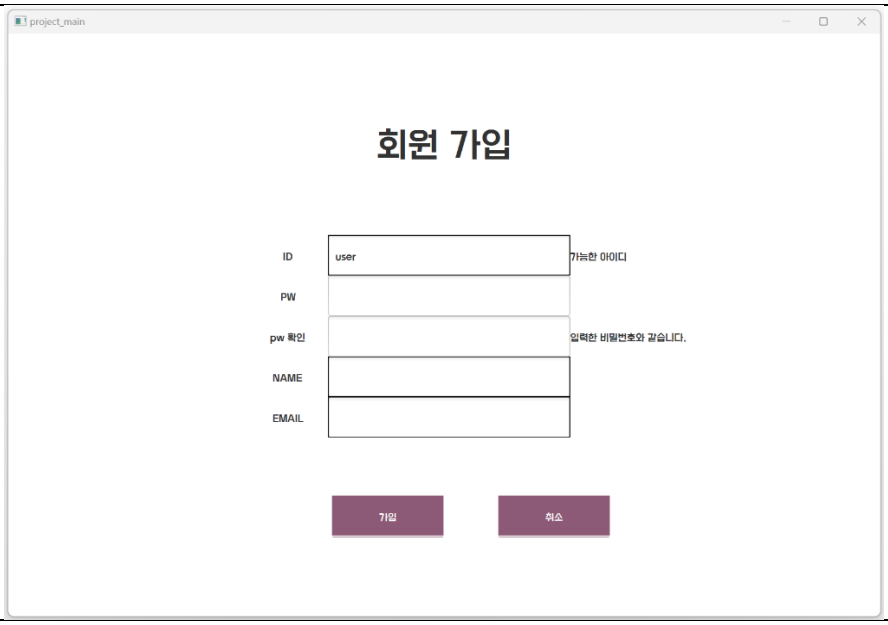
9. 스토리보드

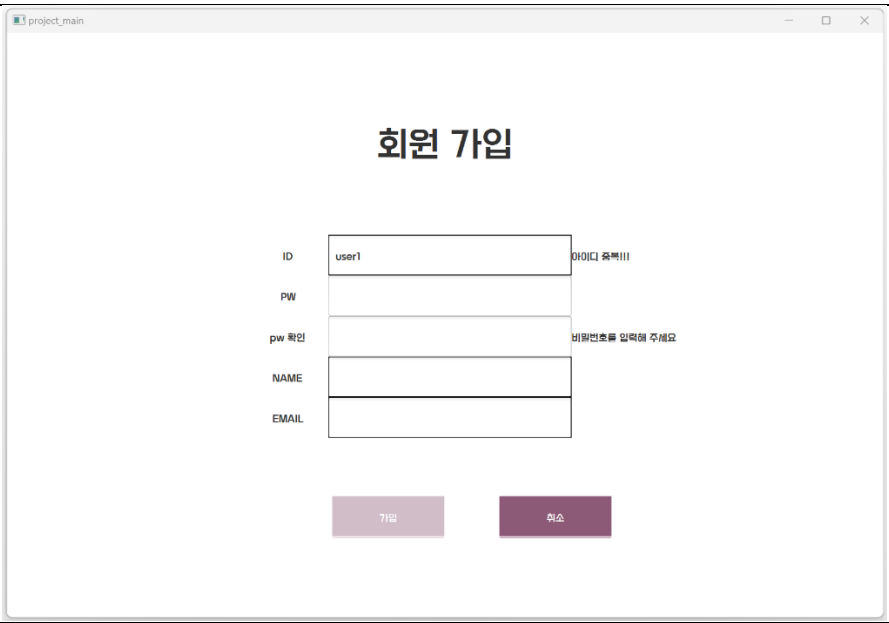
9-1. 로그인

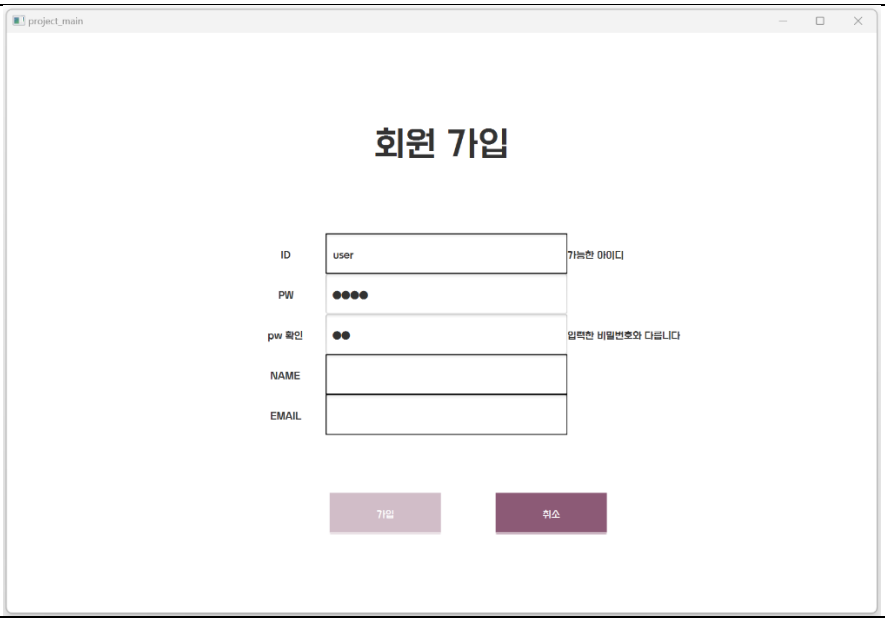
9-1-1.로그인&,회원가입 메인화면
로그인 메인 화면에서는 아이디 입력과 패스워드 입력이 가능하고 로그인 버튼과 회원가입 버튼을 눌러 로그인 또는 회원가입을 할 수 있습니다.

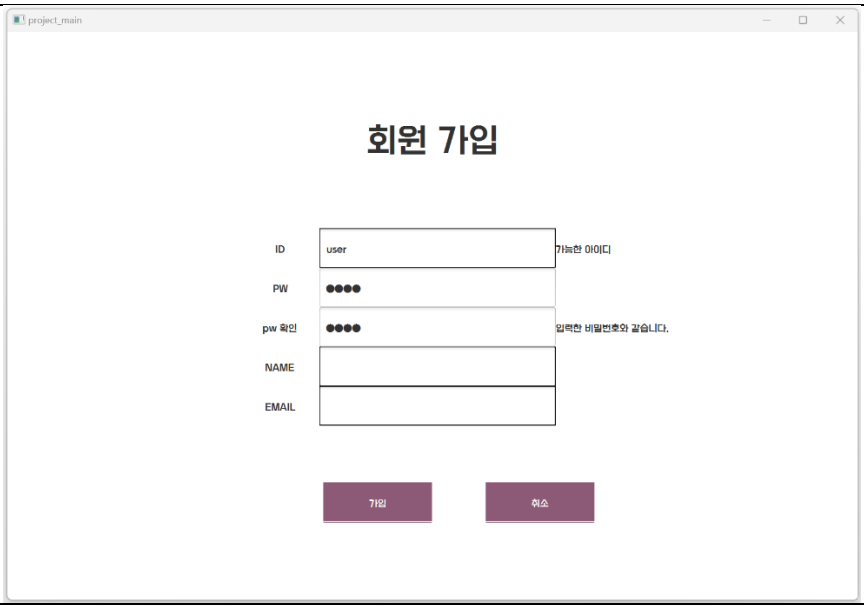
9-1-2.로그인시 아이디가 없을 경우

:로그인할 때 회원가입을 진행하지 않고 아이디를 입력할 경우 '존재하지 않는 아이디 입니다' 라는 팝업 메세지로 알 수 있습니다.

9-2.회원가입

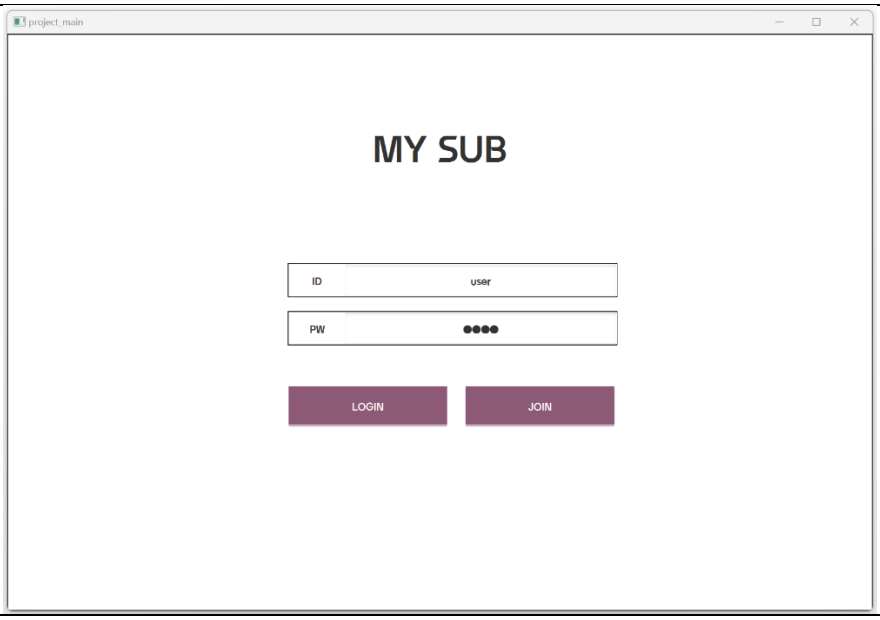
9-2-1.회원가입 시 사용가능한 아이디 일 경우

:회원가입을 할 때 중복되는 아이디가 없을 경우 '가능한 아이디' 라는 메시지로 아이디 중복이 없다는 것을 알 수 있습니다.

9-2-2.회원가입 시 입력한 아이디가 중복일 경우

:회원가입을 할 때 중복되는 아이디가 있을 경우 '아이디 중복' 라는 메시지로 아이디 중복이 있다는 것을 알 수 있습니다.

9-2-3.회원가입 시 입력한 비밀번호와 비밀번호 확인 값이 다를 경우

: :회원가입을 할 때 입력한 비밀번호와 비밀번호확인 입력창에 다른 숫자를 입력했을 경우 '입력한 비밀번호와 다릅니다' 라는 메시지로 비밀번호 재확인을 해야 한다는 것을 알 수 있습니다.

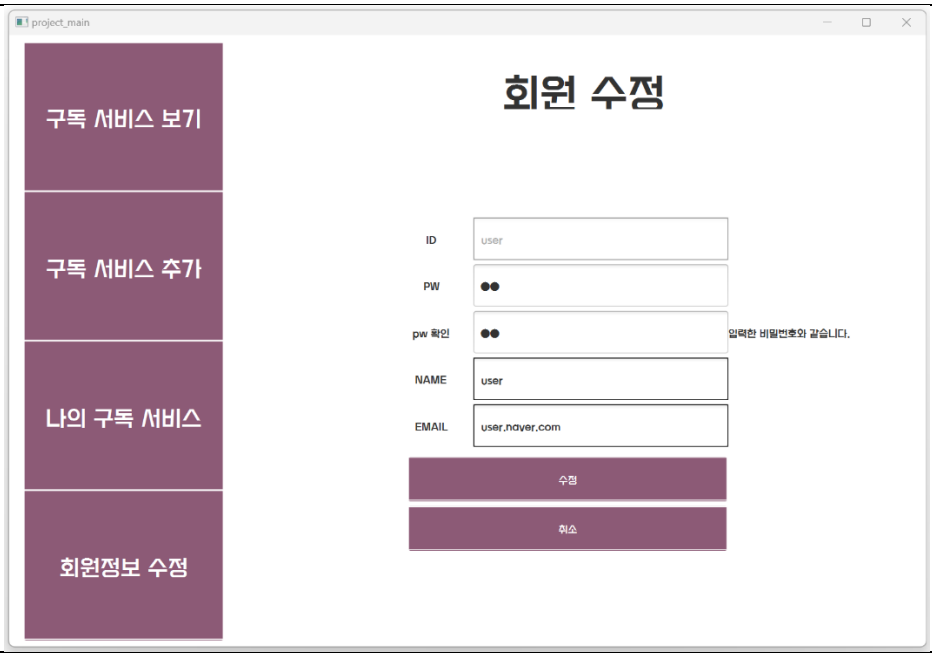
9-2-4.회원가입 시 입력한 비밀번호와 비밀번호 확인 값과 동일할 경우

:회원가입을 할 때 입력한 비밀번호와 비밀번호확인 입력창에 동일한 숫자를 입력했을 경우 '입력한 비밀번호와 같습니다' 라는 메시지로 비밀번호 확인이 되었다는 것을 알 수 있습니다.

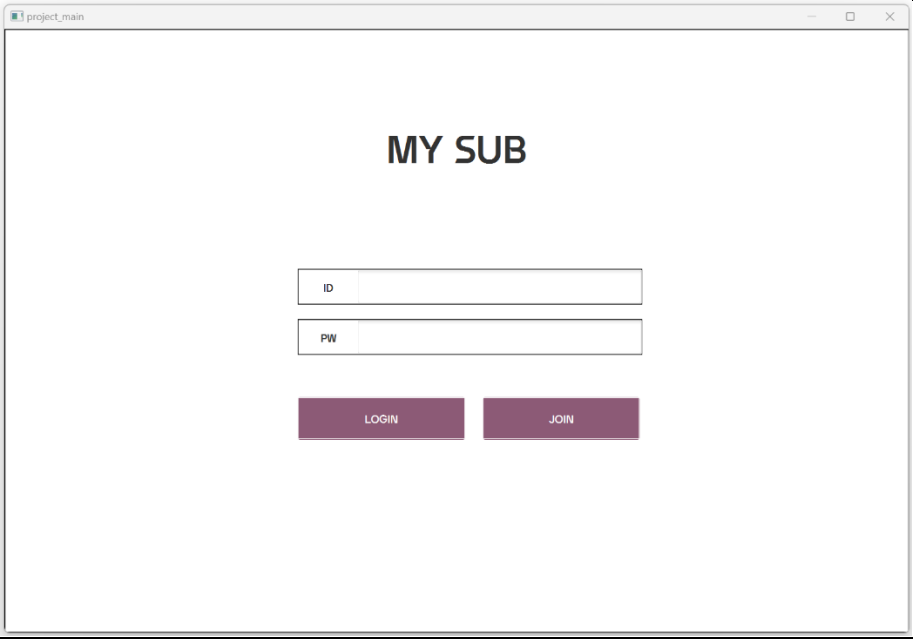
9-2-5.회원가입 필수입력 후 가입하기
:모든 조건에 충족하는 값을 입력하고 가입을 누르면 회원가입을 할 수 있습니다.

9-2-6.회원가입 완료 후 로그인

:회원가입 완료 후 아이디와 비밀번호 입력을 하면 로그인을 할 수 있습니다.

9-3.회원정보수정

9-3-1.로그인 후 메인화면에서 회원정보수정
:회원정보 수정을 원할 경우 로그인 후 메인화면에서 회원정보수정을 할 수 있습니다.

9-3-2.회원정보 수정 시 원하는 값 입력

:회원정보 수정 시 원하는 값을 입력하고 수정 버튼을 누르면 수정을 할 수 있습니다.

9-3-3.회원정보 수정 후 재로그인

:회원정보 수정을 누르면 재로그인을 위해서 로그인의 메인화면으로 돌아옵니다.

9-4. 구독서비스보기

9-4-1.구독 서비스 보기의 메인화면

:구독 서비스 보기의 메인 화면 입니다

9-4-2.구독 서비스 보기의 ALL 카테고리

:목록에 있는 모든 서비스를 볼 수 있습니다.

9-4-3. 구독 서비스 보기의 Movie 카테고리

:Movie에 해당하는 서비스만 볼 수 있습니다.

9-4-4. 구독 서비스 보기의 Music 카테고리

:Music에 해당하는 서비스만 볼 수 있습니다.

9-4-5. 구독 서비스 보기의 Game 카테고리

:Game에 해당하는 서비스만 볼 수 있습니다.

9-4-6. 구독 서비스 보기의 book 카테고리

:book에 해당하는 서비스만 볼 수 있습니다.

9-5. 구독서비스보기

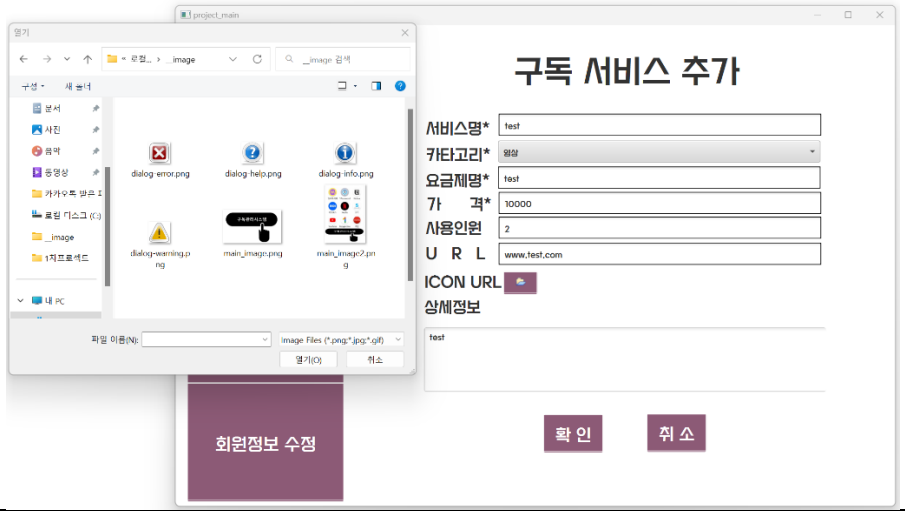
9-5-1. 구독 서비스 추가 메인화면

: 목록에 없는 서비스를 추가 할 수 있습니다

9-5-2구독 서비스 추가 입력

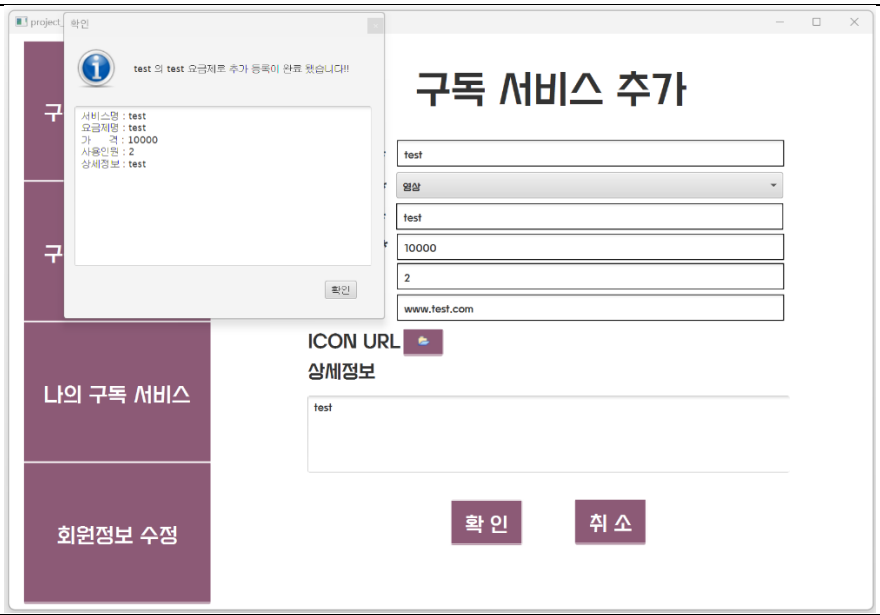
:목록에 없는 서비스에 관한 정보를 입력 할 수 있습니다.

9-5-3.추가하는 구독 서비스의 아이콘 선택



:추가하는 구독 서비스의 아이콘을 선택할 수 있습니다

9-5-4.구독 서비스 추가 입력 후 팝업창으로 확인



:서비스 추가에 관한 정보를 입력한 후 확인버튼을 누르면 팝업창으로 바로 확인할 수 있습니다.

9-5-5.구독 서비스 보기에서 추가 된 서비스 확인가능

:구독 서비스 추가에 입력한 정보는 구독 서비스 보기 목록에 추가된 것을 확인할 수 있습니다.

9-6. 나의 구독 서비스

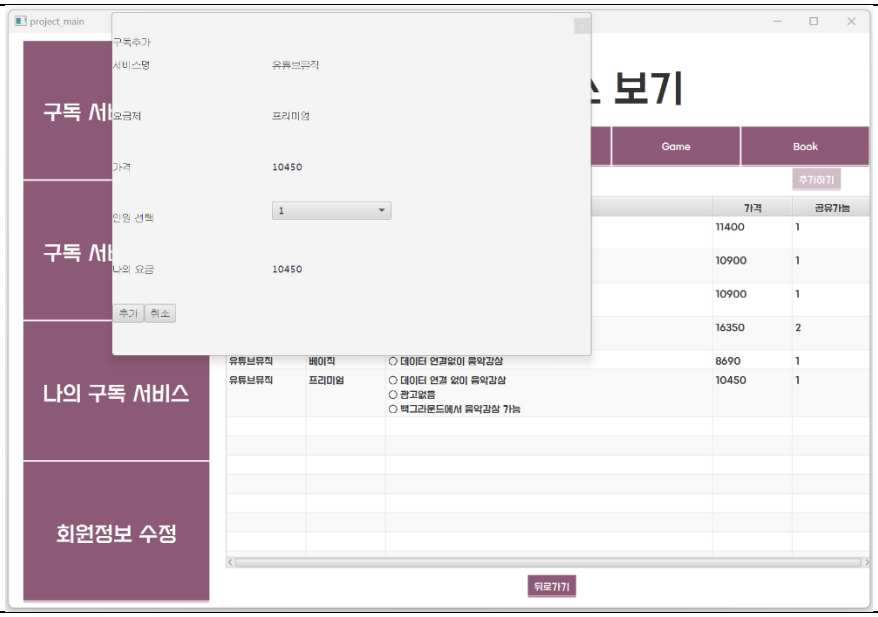
9-6-1. 나의 구독 서비스 메인화면

:현재 구독하고 있는 서비스의 현황을 볼 수 있습니다.

9-6-2.구독 서비스 보기에서 원하는 서비스 선택

:구독 서비스 보기에서 원하는 서비스를 선택하고 상단에 있는 추가하기 버튼을 누르면 나의 구독 서비스에 추가할 수 있습니다.

9-6-3.구독 서비스 보기에서 원하는 서비스 선택 상세정보 확인

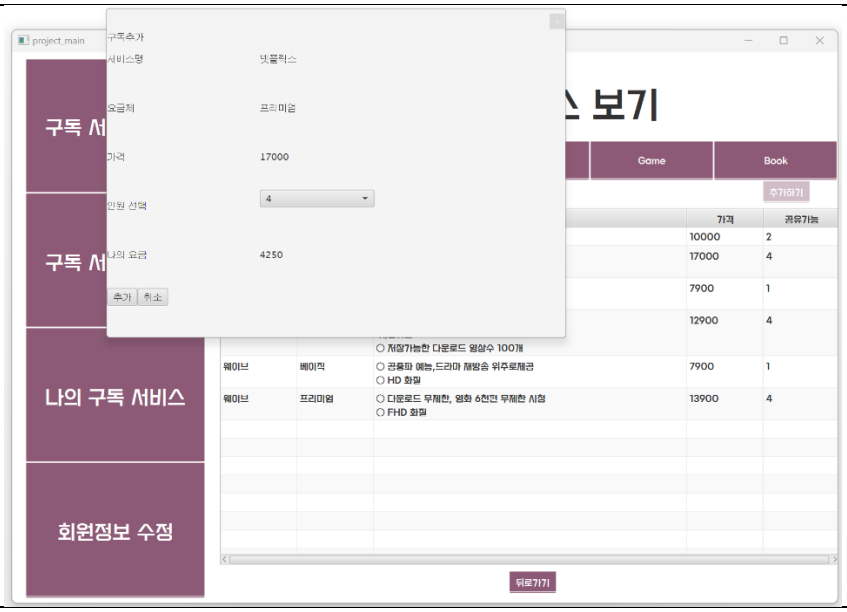


:구독 서비스 보기에서 원하는 서비스를 선택하면 상세정보를 확인할 수 있습니다.

9-6-4.나의 구독 서비스에서 추가한 서비스 확인가능

:구독 서비스 보기에서 선택하여 추가한 구독 서비스는 나의 구독 서비스에서 확인할 수 있습니다.

9-6-5.구독 서비스 보기에서 인원수에 따라 요금변경이 되는 것을 확인



:구독 서비스를 추가할 때 인원수를 변경함에 따라서 나의 요금이 변경 되는 것을 확인할 수 있습니다.

9-6-6.선택한 서비스의 총 금액 확인가능

:선택한 구독 서비스의 총금액을 하단에서 확인할 수 있습니다.

9-6-7.나의 구독 서비스에서 구독한 서비스 삭제하기

: 구독한 서비스를 삭제하고 싶은 경우 선택하여 상단의 삭제하기 버튼을눌러서 삭제할 수 있습니다.

9-6-8.구독한 서비스 삭제 후 총금액 변경 확인가능

:구독한 서비스가 삭제되면 삭제한 서비스를 제외하고 남아있는 서비스의 총금액으로 확인할 수 있습니다.

10. 테이블 명세서

테이블명	컬럼명	데이터타입	길이	PK/FK	Null 여부	Comments
CATEGORY	CATEGORY	VARCHAR2	32		Y	
PAYMENT_SYSTEM	ID	NUMBER	22	Y	N	
	NUM_PEOPLE	NUMBER	22		Y	
	INFO	VARCHAR2	512		Y	
	TIER_NAME	VARCHAR2	32		Y	
	SERVICE_NAME	VARCHAR2	32		Y	
	FEE	NUMBER	22		Y	
PAYMENT_WITH_CATEGORY	FEE	NUMBER	22	FK	Y	
	TIER_NAME	VARCHAR2	32		Y	
	SERVICE_NAME	VARCHAR2	32		Y	
	NUM_PEOPLE	NUMBER	22		Y	
	ID	NUMBER	22		N	
	SERVICE_CATEGORY	VARCHAR2	32		Y	
SERVICE_USER	USER_ID	VARCHAR2	32	PK	N	
	USER_PW	VARCHAR2	128		Y	
	USER_ROLE	VARCHAR2	8		Y	
	USER_EMAIL	VARCHAR2	64		Y	
	USER_NAME	VARCHAR2	32		Y	
SUBSCRIBE_SERVICE	SERVICE_NAME	VARCHAR2	32	PK	N	
	SERVICE_ICON_URL	VARCHAR2	256		Y	
	SERVICE_URL	VARCHAR2	256		Y	
	SERVICE_CATEGORY	VARCHAR2	32		Y	
SUBSCRIBING	ID	NUMBER	22	PK	N	
	PAYSYS_ID	NUMBER	22	FK	Y	
	USER_ID	VARCHAR2	32	FK	Y	
	NUM_OF_SHARE	NUMBER	22		Y	

테이블명	컬럼명	데이터타입	길이	PK/FK	Null 여부	Comments
V_CATELIST	FEE	NUMBER	22		Y	
	TIER_NAME	VARCHAR2	32		Y	
	NUM_PEOPLE	NUMBER	22		Y	
	SERVICE_CATEGORY	VARCHAR2	32		Y	
	SERVICE_NAME	VARCHAR2	32		Y	
	ID	NUMBER	22		N	
	INFO	VARCHAR2	512		Y	

SQL 생성문

```
--테이블 삭제
drop table service_user CASCADE CONSTRAINTS;
drop table subscribe_service CASCADE CONSTRAINTS;
--drop table category cascade CONSTRAINTS;
drop table payment_system CASCADE CONSTRAINTS;
drop table subscribing CASCADE CONSTRAINTS;

--시퀀스 삭제
drop SEQUENCE subscribe_service_id_seq;
drop SEQUENCE subscribing_id_seq;
drop SEQUENCE payment_system_id_seq;
--시퀀스 생성
create SEQUENCE subscribe_service_id_seq nocache;
create SEQUENCE subscribing_id_seq nocache;
create SEQUENCE payment_system_id_seq nocache;

--user 테이블
create table service_user(
    user_id varchar2(32) primary key, --유저 아이디
    user_pw varchar2(128),   --패스워드
    user_name varchar2(32), --이름
    user_email varchar2(64));    --이메일
```

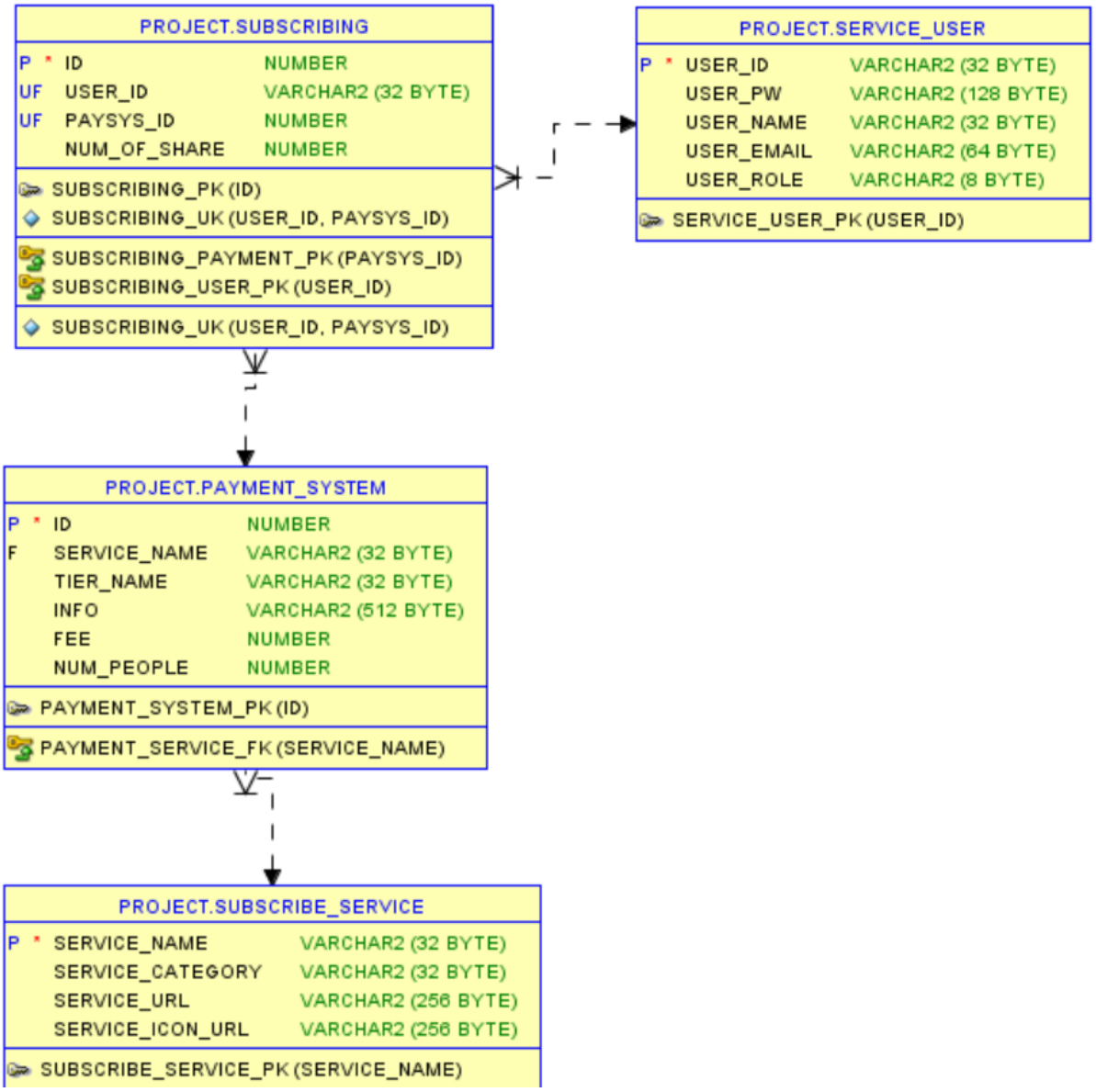
```
--구독 서비스 테이블
create table subscribe_service(
--id number default subscribe_service_id_seq.nextVal primary key
	service_name	varchar2(32) primary key, --서비스 이름 ex)넷플릭스, 왓챠
	service_category varchar(32),	--카테고리 ex)동영상, 음악
--	service_info	varchar2(512),
	service_url	varchar2(256),	--홈페이지
	service_icon_url varchar2(256)	--아이콘 url
);

--카테고리 테이블
--create table category(
--	service_name varchar2(32) primary key,
--	service_category varchar2(32),
--	CONSTRAINT name_fk FOREIGN KEY(service_name)
--	references subscribe_service(service_name) ON DELETE CASCADE
--);

--요금제 테이블
create table payment_system(
	id number default payment_system_id_seq.nextVal primary key,	--id
	service_name varchar2(32),	--구독 서비스 이름
	tier_name varchar2(32), --요금제 이름 ex) 프리미엄 , 베이직
	info varchar2(512), -- 요금제 정보
	fee number,	--요금
	num_people number,	--동시 시청인원 ex)1 , 4
	constraint payment_service_fk foreign key(service_name) references subscribe_service(service_name)
--fk 제약조건
);

create table subscribing(
	id number default subscribing_id_seq.nextVal primary key,	--id
	user_id varchar2(32),	--유저 아이디
	paysys_id number,	--요금제 id
	num_of_share number,	--공유하고 있는 인원
```

```
	constraint subscribing_payment_pk foreign key(paysys_id) references payment_system(id),
	constraint subscribing_user_pk foreign key(user_id) references service_user(user_id),
	constraint subscribing_uk unique(user_id,paysys_id)
);
alter table service_user add user_role varchar2(8) default 'USER';

create view payment_with_category
as
select ps.id as id, ps.service_name as service_name, ps.tier_name as tier_name, ps.fee as fee,
ps.num_people as num_people, ss.service_category as service_category
from payment_system ps inner join SUBSCRIBE_SERVICE ss on ps.service_name = ss.service_name;


create view category
as
select distinct service_category as category from subscribe_service;

commit;



create or replace view v_catelist
as
select a1.service_name, a1.tier_name, a1.info, a1.fee, a1.num_people, a2.service_category
from PAYMENT_SYSTEM a1 inner join SUBSCRIBE_SERVICE a2 on a1.service_name = a2.service_name;


commit;


create or replace view v_catelist as
select a1.service_name, a1.tier_name,
	a1.info, a1.fee, a1.num_people, a2.service_category,
	a1.id as id
from PAYMENT_SYSTEM a1
```

```
    inner join SUBSCRIBE_SERVICE a2
    on a1.service_name = a2.service_name;

commit;
```

11. ER다이어그램

ER 다이어그램



12. 후기

조장 임다솜

짧은 프로젝트였지만 지금까지 배운 기술들을 점검하고 부족한 부분을 채워준 뜻 깊은 프로젝트였다 생각합니다.

조장으로서 팀원들에게 큰 도움을 주지 못해 미안하고 불안한 마음도 있었지만 팀원들이 맡은 부분을 부족함 없이 수행해 주어 좋은 결과를 낼 수 있었던 것 같습니다.

조원 노설아

이번 1차 팀프로젝트를 진행하면서 프로젝트가 어떠한 방식으로 흘러가는지에 대해서 경험할 수 있었습니다. 코드의 대부분이 모르는 문법이나 형식이었기 때문에 주석을 할 때에서도 인터넷이나 책을 찾아 보면서 문법정의 위주로 이해하려고 노력하였습니다. 코드리뷰와 리허설 모든 순간이 값진 경험이었습니다. 저에게 부족한 부분을 채우기 위해서 어떠한 방향으로 나아가야 하는지에 대해서 다시한번 생각할 수 있었고 마지막으로 옆에서 함께 해준 조원들에게 정말 고생 많았고 너무너무 고마웠다고 이야기 하고 싶습니다. 3팀이 최고에요!! 마지막으로 옆에서 함께 해준 조원들에게 정말 고생 많았고 너무너무 고마웠다고 이야기 하고 싶습니다. 3팀이 최고에요!!

조원 정재웅

저의 부족한점과 구현 가능한 기능에 대해 알게 됐습니다.

특히 조원분들과 같이 협업 하여 힘든 프로젝트 과정을 잘 마무리 하면서 팀 협업에 대해 많이 배우고 느낀 것 같습니다.

조원 손일형

자 : 자그마한 몸짓으로

바 : 바스락거리는 내 코딩실력