



# 재고 관리 프로그램

**Spring Framework  
Java FX Build**

**한진수**

**han6197@naver.com**

**010 - 3744 - 6197**



# Contents

1. 프로젝트 개요

2. 프로젝트 일정

3. 개발 환경

4. 설계구조

5. 메인 페이지 작성

5-1. DB 구성

5-2. Oracle서버 DB 생성

5-3. DB 연동

5-4. 메인 페이지 View 작성

5-5. 메인 페이지 설정

5-6. 메인 페이지 DB 출력

6. 데이터 추가 기능

6-1. 데이터 추가 기능 View

6-2. 데이터 추가 기능 구현

6-3. 데이터 추가 기능 오류 처리

7. 데이터 수정 기능

7-1. 데이터 수정 기능 오류 처리

7-2. 데이터 수정 기능 구현

8. 데이터 삭제 기능

8-1. 데이터 삭제 기능 오류 처리

8-2. 데이터 삭제 기능 구현

9. 메뉴 바

9-1. DB 저장하기

9-2. 위치 별 부문 수량

9-3. 도움말

10. 프로젝트 후기

# 1. 프로젝트 개요



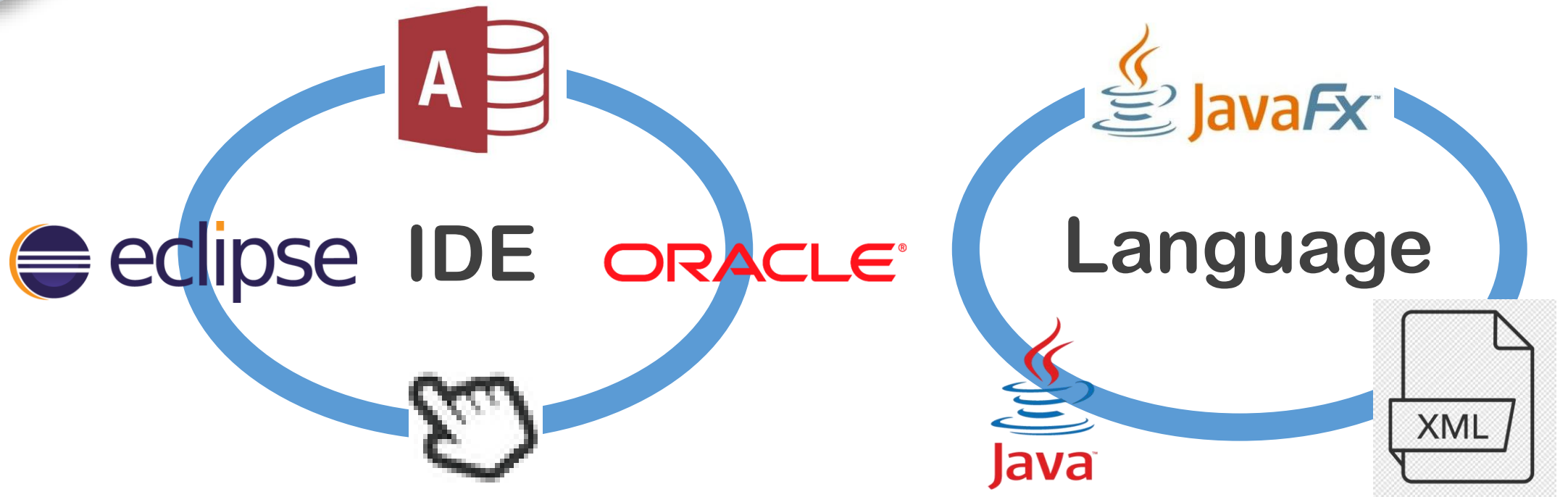
- 산업환경에서 사용하는 다양하고 많은 제품과 자재들의 부품 코드 번호와 재고 수량, 로케이션을 데이터베이스에 저장, 관리하여 보다 체계적인 재고 관리 시스템을 구축하여 효율적인 업무 활동과 정확한 재고 관리를 목적으로 한 재고 관리 시스템 체계를 구성한다.
- 사용자가 쉽게 사용할 수 있고 쉽게 접할 수 있는 프로그램 개발



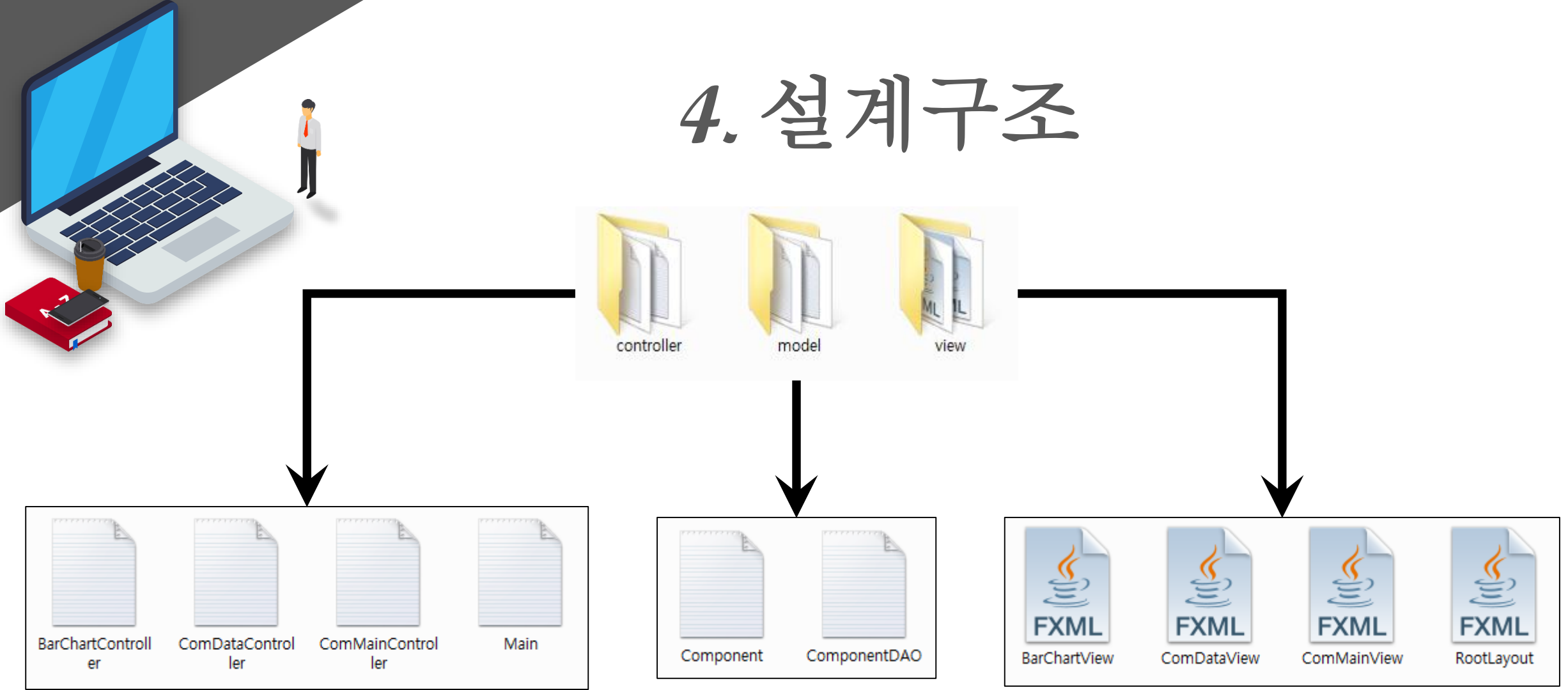
## 2. 프로젝트 일정

| 일    | 월                           | 화  | 수                   | 목  | 금           | 토   |
|------|-----------------------------|----|---------------------|----|-------------|-----|
| 7/19 | 20<br>개인 프로젝트 기획            | 21 | 22<br>DB 구상 및 DB 작성 | 23 | 24          | 25  |
| 26   | 27<br>MVC 모델 틀 잡기           | 28 | 29                  | 30 | 31<br>DB 연동 | 8/1 |
| 2    | 3<br>메뉴 바, 추가, 수정, 삭제 기능 구현 | 4  | 5                   | 6  | 7<br>통계 구현  | 8   |

### 3. 개발 환경



# 4. 설계구조



# 5-1. DB 구성

모든 Access 개체 << >>

검색...

테이블

- 고객정보
- 부품재고현황
- 영업점

| 부품코드  | 분류    | 수량   | 위치  | 추가하려면 클릭 |
|-------|-------|------|-----|----------|
| BT-01 | 배터리   | 5    | B17 |          |
| BT-02 | 배터리   | 4    | B17 |          |
| BT-03 | 배터리   | 4    | B17 |          |
| CS-01 | 코일스프링 | 6    | C6  |          |
| CS-02 | 코일스프링 | 3    | C6  |          |
| CS-03 | 코일스프링 | 7    | C7  |          |
| CS-04 | 코일스프링 | 7    | C7  |          |
| EP-01 | 배기관   | 5    | E3  |          |
| EP-02 | 배기관   | 4    | E3  |          |
| EP-03 | 배기관   | 4    | E3  |          |
| NS-01 | 나사    | 85   | N21 |          |
| NS-02 | 나사    | 1100 | N21 |          |
| NS-03 | 나사    | 135  | N22 |          |
| NS-04 | 나사    | 273  | N22 |          |
| NS-05 | 나사    | 285  | N22 |          |
| SW-01 | 소음기   | 6    | S11 |          |
| SW-02 | 소음기   | 7    | S11 |          |
| SW-03 | 소음기   | 5    | S11 |          |
| TI-01 | 타이어   | 7    | T7  |          |
| TI-02 | 타이어   | 8    | T7  |          |
| TI-03 | 타이어   | 12   | T8  |          |
| TI-04 | 타이어   | 10   | T8  |          |
| WH-01 | 휠     | 3    | W13 |          |
| WH-02 | 휠     | 7    | W13 |          |
| WH-03 | 휠     | 2    | W14 |          |
| WH-04 | 휠     | 5    | W14 |          |
| WH-05 | 휠     | 6    | W15 |          |
| WP-01 | 와이퍼   | 35   | W19 |          |
| WP-02 | 와이퍼   | 8    | W19 |          |
| WP-03 | 와이퍼   | 15   | W19 |          |
| WP-04 | 와이퍼   | 20   | W19 |          |

DB구성 및 데이터 입력

## 5-2. Oracle 서버 DB 생성

데이터 임포트 마법사 - 단계 1/5

### 데이터 미리보기

소스: 로컬 파일

파일: C:\Users\Administrator\Desktop\작업2\파일\부품재고현황.xlsx

파일 형식

☒ 헤더      행 건너뛰기(K): 0

형식(F): excel 95-2003 (xls)      ☒ 미리보기 행 제한(P): 100

파일 내용

| 부품코드 | 분류    | 수량   | 위치  |
|------|-------|------|-----|
| BT01 | 배터리   | 5    | B17 |
| BT02 | 배터리   | 4    | B17 |
| BT03 | 배터리   | 4    | B17 |
| CS01 | 코일스프링 | 6    | C6  |
| CS02 | 코일스프링 | 3    | C6  |
| CS03 | 코일스프링 | 7    | C7  |
| CS04 | 코일스프링 | 7    | C7  |
| EP01 | 배기관   | 5    | E3  |
| EP02 | 배기관   | 4    | E3  |
| EP03 | 배기관   | 4    | E3  |
| NS01 | 나사    | 85   | N21 |
| NS02 | 나사    | 1100 | N21 |
| NS03 | 나사    | 135  | N22 |
| NS04 | 나사    | 273  | N22 |
| NS05 | 나사    | 285  | N22 |

도움말(H)      < 뒤로(B)      다음(N) >      완료(F)      취소

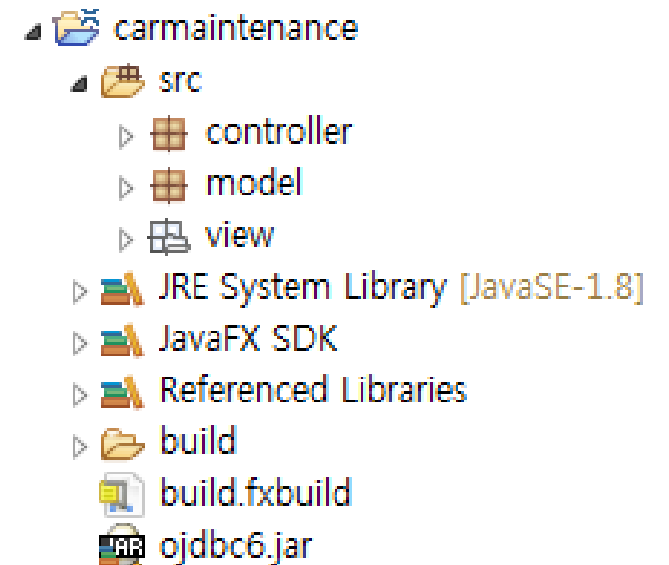
Oracle SQL Developer 외부 데이터 Import 테이블 작성



## 5-3. DB 연동

### DAO 작성 및 DB 연동 Oracle JDBC 라이브러리 추가

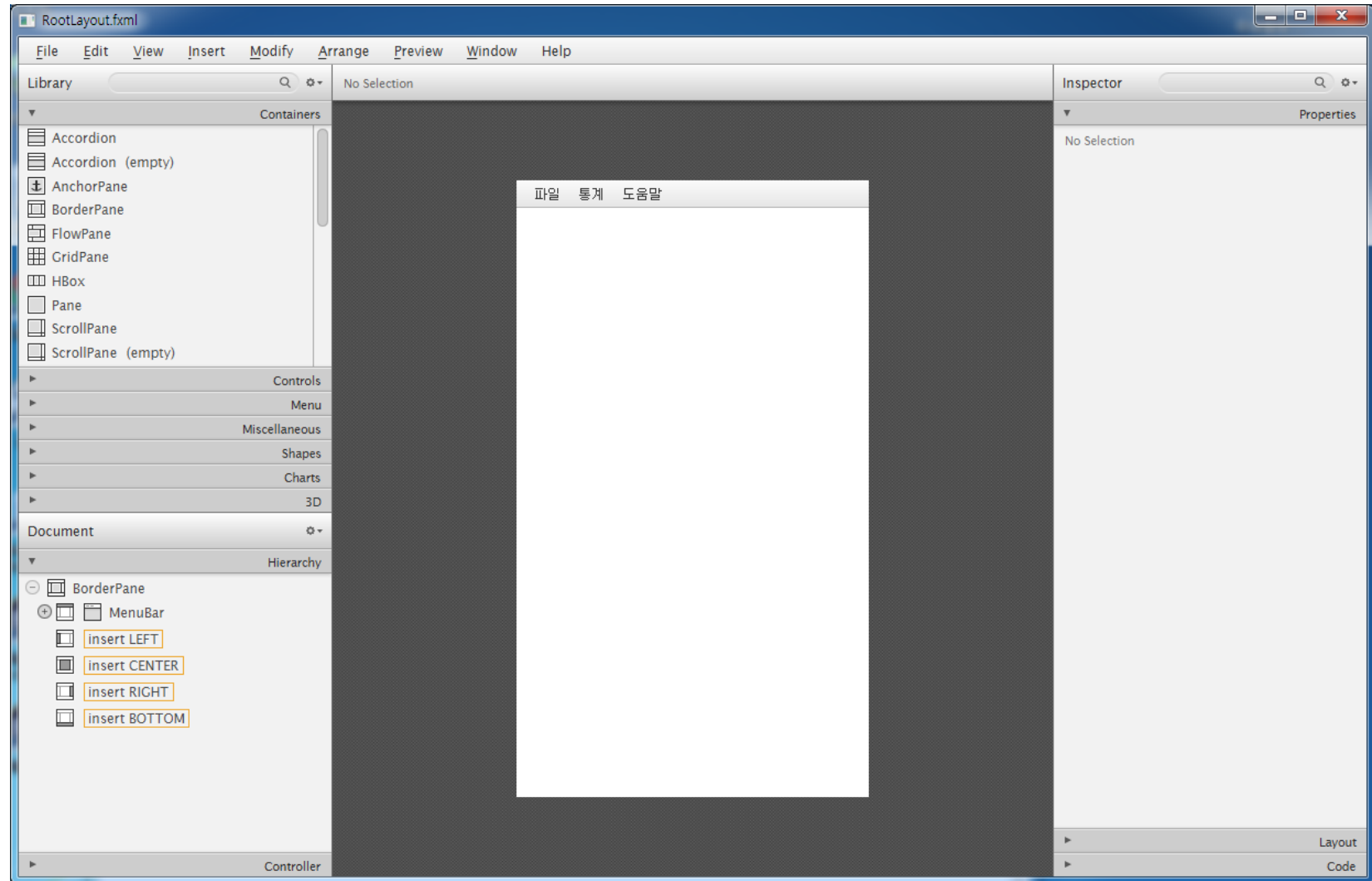
```
// DB 접속
public ComponentDAO() {
    try {
        Class.forName("oracle.jdbc.OracleDriver");
        conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
            "hr2", "12345");
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```



## 5-4. 메인 페이지 View 작성



RootLayout



## 5-4. 메인 페이지 View 작성

ComMainView



## 5-5. 메인 페이지 설정



// RootLayout 초기화

```
public void setRootLayout() {  
    try {  
        // FXML 파일을 이용해 루트 레이아웃 import  
        FXMLLoader loader = new FXMLLoader();  
        loader.setLocation(Main.class.getResource("../view/RootLayout.fxml"));  
        rootLayout = (BorderPane) loader.load();  
        // 루트 레이아웃을 포함하는 Scene을 보여줍니다.  
        Scene scene = new Scene(rootLayout);  
        primaryStage.setScene(scene);  
        primaryStage.show();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

// RootLayout 안에 Main 페이지

```
public void setComMainView() {  
    try {  
        FXMLLoader loader = new FXMLLoader();  
        loader.setLocation(Main.class.getResource("../view/ComMainView.fxml"));  
        AnchorPane comMainView = (AnchorPane) loader.load();  
        rootLayout.setCenter(comMainView);  
  
        ComMainController controller = loader.getController();  
        controller.setMain(this);  
  
        ComponentDAO componentDAO = new ComponentDAO();  
        ObservableList<Component> tempList = componentDAO.getComponentList();  
        for (int i = 0; i < tempList.size(); i++) {  
            componentList.add(tempList.get(i));  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

## 5-6. 메인 페이지 DB 출력

// DB 출력

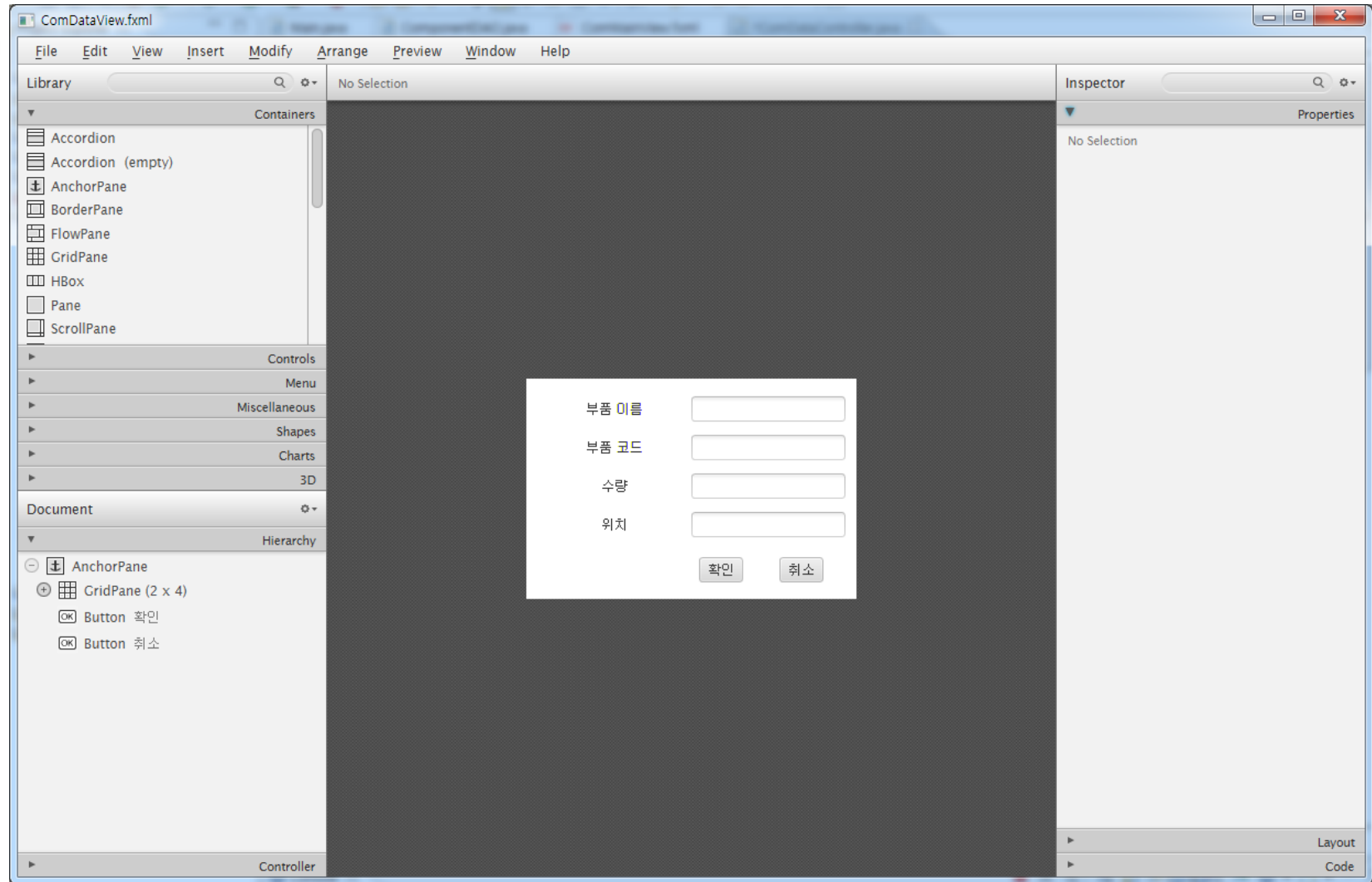
```
public ObservableList<Component> getComponentList() {  
    String SQL = "SELECT * FROM COMPONENT";  
    ObservableList<Component> componentList = FXCollections.observableArrayList();  
    try {  
        PreparedStatement pstmt = conn.prepareStatement(SQL);  
        rs = pstmt.executeQuery();  
        while (rs.next()) {  
            Component component = new Component(rs.getString("부품이름"),  
                rs.getString("부품코드"), rs.getString("수량"),  
                rs.getString("위치"));  
            componentList.add(component);  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return componentList;  
}
```



| 부품 이름 | 부품 코드 | 수량   | 위치  |
|-------|-------|------|-----|
| 코일스프링 | CS03  | 7    | C7  |
| 코일스프링 | CS04  | 7    | C7  |
| 배기관   | EP01  | 5    | E3  |
| 배기관   | EP02  | 4    | E3  |
| 배기관   | EP03  | 4    | E3  |
| 나사    | NS01  | 85   | N21 |
| 나사    | NS02  | 1100 | N21 |
| 나사    | NS03  | 135  | N22 |
| 나사    | NS04  | 273  | N22 |
| 나사    | NS05  | 285  | N22 |
| 소음기   | SW01  | 6    | S11 |
| 소음기   | SW02  | 7    | S11 |
| 소음기   | SW03  | 5    | S11 |
| 타이어   | TI01  | 7    | T7  |
| 타이어   | TI02  | 8    | T7  |
| 타이어   | TI03  | 12   | T8  |
| 타이어   | TI04  | 10   | T8  |

# 6-1. 데이터 추가 기능 View

ComDataView



## 6-2. 데이터 추가 기능 구현

```
public int setComponentDataView(Component component) {  
    try {  
        FXMLLoader loader = new FXMLLoader();  
        loader.setLocation(Main.class.getResource("../view/ComDataView.fxml"));  
        AnchorPane page = (AnchorPane) loader.load();  
  
        Stage dialogStage = new Stage();  
        dialogStage.setTitle("부품 추가");  
        dialogStage.initModality(Modality.WINDOW_MODAL);  
        dialogStage.initOwner(primaryStage);  
        Scene scene = new Scene(page);  
        dialogStage.setScene(scene);  
  
        ComDataController controller = loader.getController();  
        controller.setDialogStage(dialogStage);  
        controller.setComponent(component);  
  
        dialogStage.showAndWait();  
        return controller.getReturnValue();  
    } catch (IOException e) {  
        e.printStackTrace();  
        return 0;  
    }  
}
```

부품 추가 페이지 열기





## 6-2. 데이터 추가 기능 구현

@FXML

```
private void addAction() {  
    Component component = new Component("", "", "", "");  
    int returnValue = main.setComponentDataView(component);  
    if (returnValue == 1) {  
        main.getComponentList().add(component);  
    }  
}
```

추가(addAction) 버튼  
클릭 시 List에 데이터 추가

}

```
public void setComponent(Component component) {  
    this.component = component;  
    cnameField.setText(component.getCname());  
    cnumberField.setText(component.getCnumber());  
    ccountField.setText(component.getCcount());  
    clocField.setText(component.getCloc());  
}
```

데이터 입력 받고  
새로운 데이터 생성

}

@FXML

```
private void confirmAction() {  
    if (valid()) {  
        component.setName(cnameField.getText());  
        component.setCnumber(cnumberField.getText());  
        component.setCcount(ccountField.getText());  
        component.setCloc(clocField.getText());  
        returnValue = 1;  
        dialogStage.close();  
    }  
}
```

확인 버튼 클릭 시  
confirmAction 실행

}

}

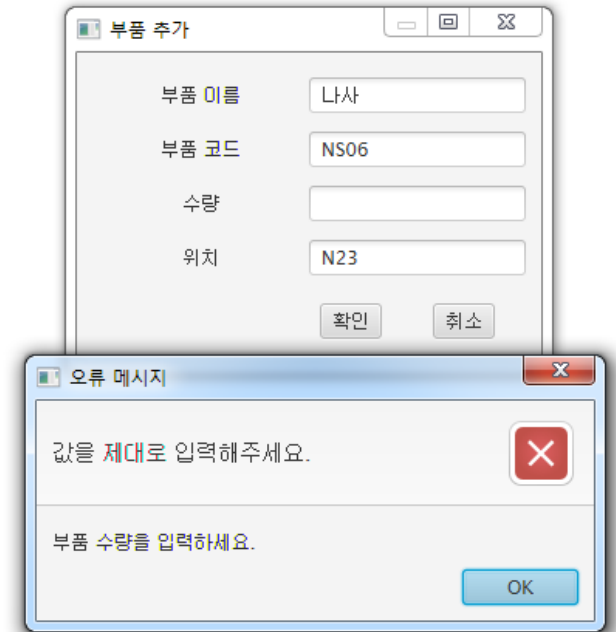






## 6-3. 데이터 추가 기능 오류 처리

```
private boolean valid() {  
    String errorMessage = "";  
    if (cnameField.getText() == null || cnameField.getText().equals("")) {  
        errorMessage += "부품 이름을 입력하세요.\n";  
    }  
    if (cnumberField.getText() == null || cnumberField.getText().equals("")) {  
        errorMessage += "부품 코드를 입력하세요.\n";  
    }  
    if (ccountField.getText() == null || ccountField.getText().equals("")) {  
        errorMessage += "부품 수량을 입력하세요.\n";  
    }  
    if (errorMessage.equals("")) {  
        return true;  
    } else {  
        Alert alert = new Alert(AlertType.ERROR);  
        alert.initOwner(dialogStage);  
        alert.setTitle("오류 메시지");  
        alert.setHeaderText("값을 제대로 입력해주세요.");  
        alert.setContentText(errorMessage);  
        alert.showAndWait();  
        return false;  
    }  
}
```



공백 입력 시 데이터 추가 기능이 실행되지 않고 에러 메시지 출력

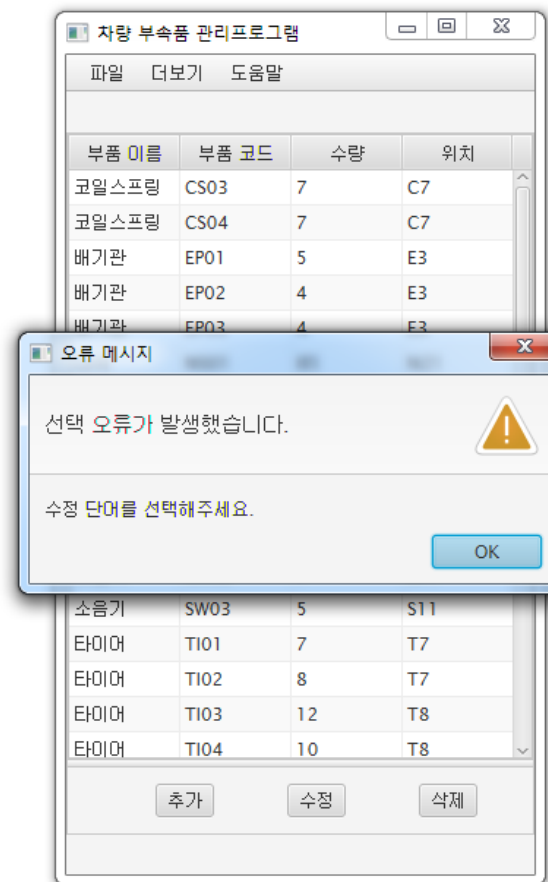


# 7-1. 데이터 수정 기능 오류 처리

@FXML

```
private void editAction() {  
    Component component = comTable.getSelectionModel().getSelectedItem();  
    if (component != null) {  
        main.setComponentDataView(component);  
    } else {  
        Alert alert = new Alert(AlertType.WARNING);  
        alert.initOwner(main.getPrimaryStage());  
        alert.setTitle("오류 메시지");  
        alert.setHeaderText("선택 오류가 발생했습니다.");  
        alert.setContentText("수정 단어를 선택해 주세요.");  
        alert.showAndWait();  
    }  
}
```

수정할 단어가 선택되지 않았다면 오류 메시지 출력



## 7-2. 데이터 수정 기능 구현

차량 부품품 관리프로그램

파일 더보기 도움말

| 부품 이름 | 부품 코드 | 수량 | 위치 |
|-------|-------|----|----|
| 타이머   | TI02  | 8  | T7 |

부품 추가

부품 이름:

부품 코드:

수량:

위치:

확인 취소

|       |      |    |     |
|-------|------|----|-----|
| 와이퍼   | WP04 | 20 | W19 |
| 배터리   | BT04 | 3  | B18 |
| 배터리   | BT01 | 5  | B17 |
| 배터리   | BT02 | 4  | B17 |
| 코일스프링 | CS01 | 6  | C6  |
| 코일스프링 | CS02 | 3  | C6  |

추가 수정 삭제

수정

차량 부품품 관리프로그램

파일 더보기 도움말

| 부품 이름 | 부품 코드 | 수량 | 위치  |
|-------|-------|----|-----|
| 타이머   | TI02  | 8  | T7  |
| 타이머   | TI03  | 12 | T8  |
| 타이머   | TI04  | 10 | T8  |
| 휠     | WH01  | 3  | W13 |
| 휠     | WH02  | 7  | W13 |
| 휠     | WH03  | 2  | W14 |
| 휠     | WH04  | 5  | W14 |
| 휠     | WH05  | 6  | W15 |
| 와이퍼   | WP01  | 35 | W19 |
| 와이퍼   | WP02  | 8  | W19 |
| 와이퍼   | WP03  | 15 | W19 |
| 와이퍼   | WP04  | 20 | W19 |
| 배터리   | BT03  | 4  | B18 |
| 배터리   | BT01  | 5  | B17 |
| 배터리   | BT02  | 4  | B17 |
| 코일스프링 | CS01  | 6  | C6  |
| 코일스프링 | CS02  | 3  | C6  |

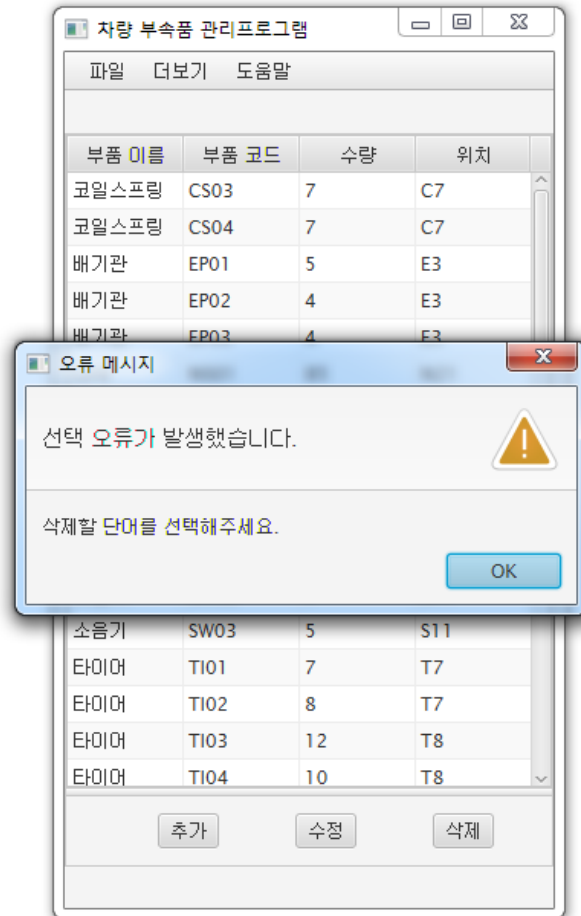
추가 수정 삭제

# 8-1. 데이터 삭제 기능 오류 처리

@FXML

```
private void deleteAction() {  
    int selectedIndex = comTable.getSelectionModel().getSelectedIndex();  
    if (selectedIndex >= 0) {  
        comTable.getItems().remove(selectedIndex);  
    } else {  
        Alert alert = new Alert(AlertType.WARNING);  
        alert.initOwner(main.getPrimaryStage());  
        alert.setTitle("오류 메시지");  
        alert.setHeaderText("선택 오류가 발생했습니다.");  
        alert.setContentText("삭제할 단어를 선택해주세요.");  
        alert.showAndWait();  
    }  
}
```

삭제할 단어가 선택되지 않았다면 오류 메시지 출력



## 8-2. 데이터 삭제 기능 구현

차량 부품품 관리프로그램

파일 더보기 도움말

| 부품 이름 | 부품 코드 | 수량 | 위치  |
|-------|-------|----|-----|
| 타이머   | TI02  | 8  | T7  |
| 타이머   | TI03  | 12 | T8  |
| 타이머   | TI04  | 10 | T8  |
| 휠     | WH01  | 3  | W13 |
| 휠     | WH02  | 7  | W13 |
| 휠     | WH03  | 2  | W14 |
| 휠     | WH04  | 5  | W14 |
| 휠     | WH05  | 6  | W15 |
| 와이퍼   | WP01  | 35 | W19 |
| 와이퍼   | WP02  | 8  | W19 |
| 와이퍼   | WP03  | 15 | W19 |
| 와이퍼   | WP04  | 20 | W19 |
| 배터리   | BT04  | 3  | B18 |
| 배터리   | BT01  | 5  | B17 |
| 배터리   | BT02  | 4  | B17 |
| 코일스프링 | CS01  | 6  | C6  |
| 코일스프링 | CS02  | 3  | C6  |

추가 수정 삭제

삭제

차량 부품품 관리프로그램

파일 더보기 도움말

| 부품 이름 | 부품 코드 | 수량 | 위치  |
|-------|-------|----|-----|
| 타이머   | TI01  | 7  | T7  |
| 타이머   | TI02  | 8  | T7  |
| 타이머   | TI03  | 12 | T8  |
| 타이머   | TI04  | 10 | T8  |
| 휠     | WH01  | 3  | W13 |
| 휠     | WH02  | 7  | W13 |
| 휠     | WH03  | 2  | W14 |
| 휠     | WH04  | 5  | W14 |
| 휠     | WH05  | 6  | W15 |
| 와이퍼   | WP01  | 35 | W19 |
| 와이퍼   | WP02  | 8  | W19 |
| 와이퍼   | WP03  | 15 | W19 |
| 와이퍼   | WP04  | 20 | W19 |
| 배터리   | BT04  | 3  | B18 |
| 배터리   | BT01  | 5  | B17 |
| 배터리   | BT02  | 4  | B17 |
| 코일스프링 | CS01  | 6  | C6  |

추가 수정 삭제



# 9-1. DB 저장하기

```
// DB에 저장
public int saveComponentList(ObservableList<Component> componentList) {
    if (deleteComponentList() == -1) {
        return -1;
    }
    if (insertComponentList(componentList) == -1) {
        return -1;
    }
    return 1;
}
```

DAO – Insert, Delete 된 데이터 DB 서버에 적용

```
@FXML
private void saveAction() {
    ComponentDAO componentDAO = new ComponentDAO();
    int result = componentDAO.saveComponentList(componentList);
    if (result == 1) {
        Alert alert = new Alert(AlertType.CONFIRMATION);
        alert.initOwner(primaryStage);
        alert.setTitle("성공 메시지");
        alert.setHeaderText("성공적으로 수행했습니다.");
        alert.setContentText("데이터베이스에 성공적으로 접근했습니다.");
        alert.showAndWait();
    } else {
        Alert alert = new Alert(AlertType.ERROR);
        alert.initOwner(primaryStage);
        alert.setTitle("오류 메시지");
        alert.setHeaderText("오류가 발생했습니다.");
        alert.setContentText("데이터베이스에 오류가 발생했습니다.");
        alert.showAndWait();
    }
}
```

Main – 저장하기 버튼 클릭 후 DB 서버에 저장

# 9-1. DB 저장하기

## DB Server

차량 부품품 관리프로그램

파일 더보기 도움말

저장하기

종료하기

|       | 부품 코드 | 수량 | 위치  |
|-------|-------|----|-----|
| 타이어   | TI01  | 7  | T7  |
| 타이어   | TI02  | 8  | T7  |
| 타이어   | TI03  | 12 | T8  |
| 타이어   | TI04  | 10 | T8  |
| 휠     | WH01  | 3  | W13 |
| 휠     | WH02  | 7  | W13 |
| 휠     | WH03  | 2  | W14 |
| 휠     | WH04  | 5  | W14 |
| 휠     | WH05  | 6  | W15 |
| 와이퍼   | WP01  | 35 | W19 |
| 와이퍼   | WP02  | 8  | W19 |
| 와이퍼   | WP03  | 15 | W19 |
| 와이퍼   | WP04  | 20 | W19 |
| 배터리   | BT04  | 3  | B18 |
| 배터리   | BT01  | 5  | B17 |
| 배터리   | BT02  | 4  | B17 |
| 코일스프링 | CS01  | 6  | C6  |

추가 수정 삭제

차량 부품품 관리프로그램

파일 더보기 도움말

| 부품 이름 | 부품 코드 | 수량 | 위치 |
|-------|-------|----|----|
| 타이어   | TI01  | 7  | T7 |

성공 메시지

성공적으로 수행했습니다.

데이터베이스에 성공적으로 접근했습니다.

OK 취소


| 와이퍼   | WP01 | 35 | W19 |
|-------|------|----|-----|
| 와이퍼   | WP02 | 8  | W19 |
| 와이퍼   | WP03 | 15 | W19 |
| 와이퍼   | WP04 | 20 | W19 |
| 배터리   | BT04 | 3  | B18 |
| 배터리   | BT01 | 5  | B17 |
| 배터리   | BT02 | 4  | B17 |
| 코일스프링 | CS01 | 6  | C6  |

추가 수정 삭제

|    | 부품코드 | 부품이름  | 수량   | 위치  |
|----|------|-------|------|-----|
| 1  | NS03 | 나사    | 135  | N22 |
| 2  | NS05 | 나사    | 285  | N22 |
| 3  | NS04 | 나사    | 273  | N22 |
| 4  | NS02 | 나사    | 1100 | N21 |
| 5  | NS01 | 나사    | 85   | N21 |
| 6  | EP02 | 배기관   | 4    | E3  |
| 7  | EP01 | 배기관   | 5    | E3  |
| 8  | EP03 | 배기관   | 4    | E3  |
| 9  | BT04 | 배터리   | 3    | B18 |
| 10 | BT02 | 배터리   | 4    | B17 |
| 11 | BT01 | 배터리   | 5    | B17 |
| 12 | SW03 | 소음기   | 5    | S11 |
| 13 | SW02 | 소음기   | 7    | S11 |
| 14 | SW01 | 소음기   | 6    | S11 |
| 15 | WP04 | 와이퍼   | 20   | W19 |
| 16 | WP03 | 와이퍼   | 15   | W19 |
| 17 | WP02 | 와이퍼   | 8    | W19 |
| 18 | WP01 | 와이퍼   | 35   | W19 |
| 19 | CS03 | 코일스프링 | 7    | C7  |
| 20 | CS04 | 코일스프링 | 7    | C7  |
| 21 | CS01 | 코일스프링 | 6    | C6  |
| 22 | TI02 | 타이어   | 8    | T7  |
| 23 | TI01 | 타이어   | 7    | T7  |
| 24 | TI03 | 타이어   | 12   | T8  |
| 25 | TI04 | 타이어   | 10   | T8  |
| 26 | WH04 | 휠     | 5    | W14 |
| 27 | WH03 | 휠     | 2    | W14 |
| 28 | WH05 | 휠     | 6    | W15 |
| 29 | WH01 | 휠     | 3    | W13 |
| 30 | WH02 | 휠     | 7    | W13 |

저장하기를 실행해야 변경내용이 DB 서버에 적용

# 9-1. DB 저장하기



차량 부품품 관리프로그램

파일 더보기 도움말

저장하기  
종료하기

|       | 부품 코드 | 수량 | 위치  |
|-------|-------|----|-----|
| 타미너   | TI04  | 10 | T8  |
| 휠     | WH01  | 3  | W13 |
| 휠     | WH02  | 7  | W13 |
| 휠     | WH03  | 2  | W14 |
| 휠     | WH04  | 5  | W14 |
| 휠     | WH05  | 6  | W15 |
| 와이퍼   | WP01  | 35 | W19 |
| 와이퍼   | WP02  | 8  | W19 |
| 와이퍼   | WP03  | 15 | W19 |
| 와이퍼   | WP04  | 20 | W19 |
| 배터리   | BT04  | 3  | B18 |
| 배터리   | BT01  | 5  | B17 |
| 배터리   | BT02  | 4  | B17 |
| 코일스프링 | CS01  | 6  | C6  |
| 코일스프링 | CS03  | 7  | C7  |
| 코일스프링 | CS04  | 7  | C7  |
| 코일스프링 | CS01  | 5  | C6  |

추가 수정 삭제

차량 부품품 관리프로그램

파일 더보기 도움말

| 부품 이름 | 부품 코드 | 수량 | 위치  |
|-------|-------|----|-----|
| 타미너   | TI04  | 10 | T8  |
| 휠     | WH01  | 3  | W13 |
| 휠     | WH02  | 7  | W13 |

오류 메시지

오류가 발생했습니다.

데이터베이스에 오류가 발생했습니다.

OK

|       |      |   |     |
|-------|------|---|-----|
| 배터리   | BT01 | 5 | B17 |
| 배터리   | BT02 | 4 | B17 |
| 코일스프링 | CS01 | 6 | C6  |
| 코일스프링 | CS03 | 7 | C7  |
| 코일스프링 | CS04 | 7 | C7  |
| 코일스프링 | CS01 | 5 | C6  |


추가 수정 삭제

프로그램에서 데이터 추가가  
되었더라도  
제약조건을 지키지 않은 데이  
터가 입력되었으면  
정상적인 실행이 되지 않는다.

Primary키인 부품 코드에서  
CS01 중복 입력되었음.



## 9-2. 위치 별 부문 수량



```
public void setComponentList(List<Component> componentList) {  
    int[] counters = new int[26];  
    for (Component component : componentList) {  
        char character = component.getCnumber().charAt(0);  
        counters[character - 65]++;  
    }  
  
    XYChart.Series<String, Integer> series = new XYChart.Series<>();  
  
    for (int i = 0; i < counters.length; i++) {  
        series.getData().add(new XYChart.Data<>(firstCharacter.get(i) + "", counters[i]));  
    }  
  
    barChart.getData().add(series);  
}
```

부품 코드 번호(Cnumber)의 글자 배열의 첫 글자를  
그룹별로 Location의 종류 수량을 Count

## 9-2. 위치 별 부문 수량

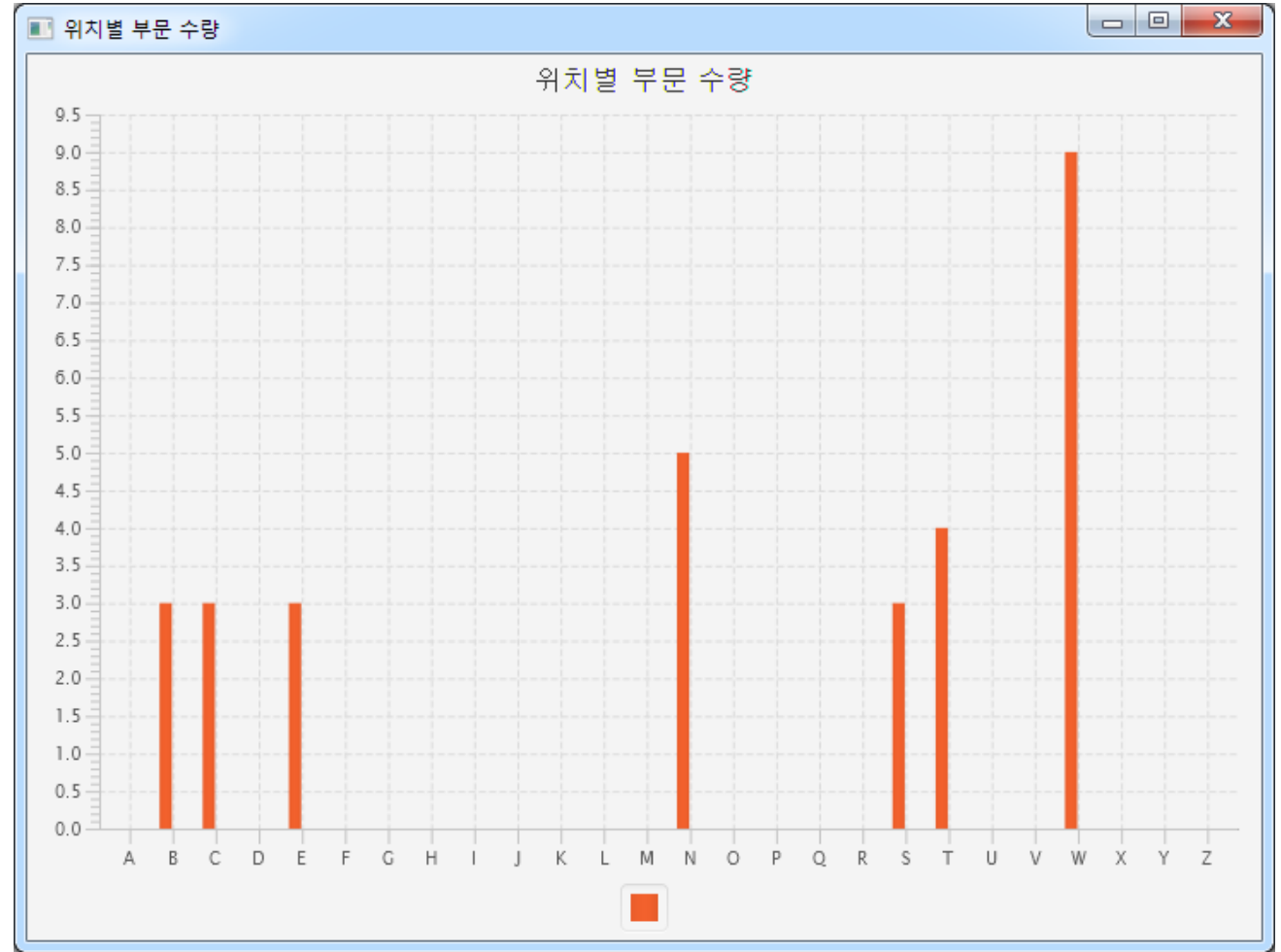
자랑 부속품 관리프로그램

파일 더보기 도움말

위치별 부문 수량

| 부품 이름 | 부품 코드 | 수량   | 위치  |
|-------|-------|------|-----|
| 배기관   | EP01  | 5    | E3  |
| 배기관   | EP02  | 4    | E3  |
| 배기관   | EP03  | 4    | E3  |
| 나사    | NS01  | 85   | N21 |
| 나사    | NS02  | 1100 | N21 |
| 나사    | NS03  | 135  | N22 |
| 나사    | NS04  | 273  | N22 |
| 나사    | NS05  | 285  | N22 |
| 소음기   | SW01  | 6    | S11 |
| 소음기   | SW02  | 7    | S11 |
| 소음기   | SW03  | 5    | S11 |
| 타이어   | TI01  | 7    | T7  |
| 타이어   | TI02  | 8    | T7  |
| 타이어   | TI03  | 12   | T8  |
| 타이어   | TI04  | 10   | T8  |
| 휠     | WH01  | 3    | W13 |
| 휠     | WH02  | 7    | W13 |

추가 수정 삭제



Location별로 저장되어있는 부품의 종류 수량을 막대그래프로 시각화

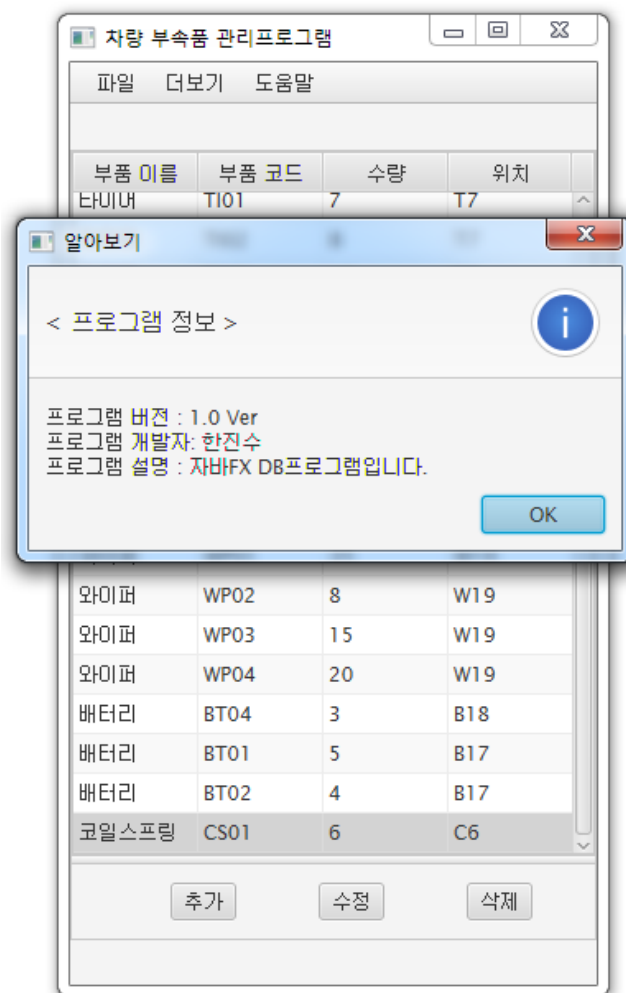
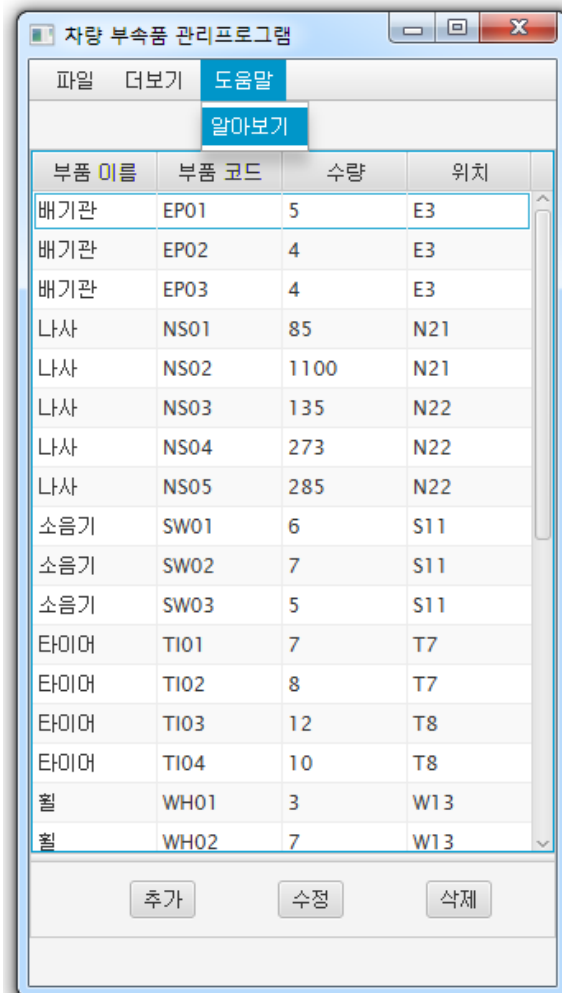


## 9-3. 도움말

@FXML

```
private void aboutAction() {
    Alert alert = new Alert(AlertType.INFORMATION);
    alert.setTitle("알아보기");
    alert.setHeaderText("< 프로그램 정보 >");
    alert.setContentText("프로그램 버전 : 1.0 Ver\n" +
        "프로그램 개발자 : 한진수\n" +
        "프로그램 설명 : 자바FX DB프로그램입니다.");
    alert.showAndWait();
}
```

도움말 기능으로 프로그램의  
기본 정보를 알려준다.





## 10. 프로젝트 후기

재고 관리 프로그램 개발을 진행하면서 다양한 생각이 들었다.

프로그램을 더 개발하여 부품이 사용된 이력이 저장되거나 부품을 사용하면 데이터베이스에 입력되지 않아도 자동으로 개수가 차감되는 기능이나 특정 개수 미만으로 내려가면 자동으로 업체에게 전달이 되어 발주까지 되는 시스템까지 개발되면 유용하겠다 등 다양한 기능들을 추가, 연동해볼 수 있겠다는 생각들이 들면서 직접 개발을 해본다는 것이 장의성이나 다음에 프로그램 개발을 할 때에도 많은 도움이 될 거 같다는 생각이 들었다.

그리고 프로젝트를 FXBuilder를 사용하여 exe실행파일로 만들어도 보고 실행파일로 이상 없이 구동되는 것도 확인해보면서 요즘 PDA 기기나 핸드폰, 스마트 기기로도 업무를 많이 보는 시대에 애플리케이션으로 개발을 해서 다양한 플랫폼으로 사용이 활성화 될 수 있도록 해도 좋겠다는 생각도 들었다.



해당 프로젝트 소스코드는 아래  
주소에서 확인 가능합니다.

<https://github.com/han6197/JavaFX>