https://github.com/lixy1979/java_week_07 Coding assginment
https://github.com/lixy1979/java_week_09 Other assginment
https://youtu.be/IIwZ7Qfx8Lg

ProjectsAPP.java

```
/**
 *
 */
package projects;

import java.math.BigDecimal;
import java.util.List;
import java.util.Objects;
import java.util.Scanner;
import projects.entity.Project;
import projects.exception.DbException;
import projects.service.ProjectService;

/**
 * @author lixy4
 *
 */
public class ProjectsApp {
       private Scanner scanner = new Scanner(System.in);
    private ProjectService projectService = new ProjectService();



       // @formatter:off
       private List<String> operations = List.of(
                     "1) Add a project"
              );
       // @formatter:on
       /**
        * @param args
        */
       public static void main(String[] args) {
       new ProjectsApp().processUserSelections();
       }

       private void processUserSelections() {
       boolean done = false;

       while(!done) {
              try {
                     int selection = getUserSelection();

                     switch(selection) {
                     case -1:
                     done = exitMenu();
                     break;

                     case 1:
                            createProject();
                            break;


                    default:
                        System.out.println("\n" + selection + " is not a valid
selection. Try again.");
                        break;
                        }
                }
                catch(Exception e) {
                        System.out.println("\nError: " + e + "Try again.");
                }
          }
        }

        private void createProject() {
                String projectName = getStringInput("Enter the project name");
                BigDecimal estimatedHours = getDecimalInput("Enter the estimated
hours");
                BigDecimal actualHours = getDecimalInput("Enter the actual hours");
                Integer difficulty = getIntInput("Enter the project difficulty (1-
5)");
                String notes = getStringInput("Enter the project notes");

                Project project = new Project();

                project.setProjectName(projectName);
                project.setEstimatedHours(estimatedHours);
                project.setActualHours(actualHours);
                project.setDifficulty(difficulty);
                project.setNotes(notes);

                Project dbProject = projectService.addProject(project);
                System.out.println("You have successfully created project: " +
dbProject);
        }

        private BigDecimal getDecimalInput(String prompt) {
         String input = getStringInput(prompt);

                if(Objects.isNull(input)) {
                        return null;
                }

                try {
                        return new BigDecimal(input).setScale(2);
                }
                catch(NumberFormatException e) {
                        throw new DbException(input + "is not a valid decimal
number.");
                }
        }
```

```
	private boolean exitMenu() {
		System.out.println("Exiting the menu.");
		return true;
	}

	private int getUserSelection() {
		printOperations();

		Integer input = getIntInput("Enter a menu selection");
		return Objects.isNull(input) ? -1 : input;
	}

	private Integer getIntInput(String prompt) {
		String input = getStringInput(prompt);

		if(Objects.isNull(input)) {
			return null;
		}

		try {
			return Integer.valueOf(input);
		}
		catch(NumberFormatException e) {
			throw new DbException(input + "is not a valid number.");
		}
	}

	private String getStringInput(String prompt) {
		System.out.print(prompt + ": ");
		String input = scanner.nextLine();

		return input.isBlank() ? null : input.trim();
	}

	private void printOperations() {
		System.out.println("\nThese are the available selections. Press the 
Enter key to quit: ");
		operations.forEach(line -> System.out.println("   " + line));
	}
}
```

ProjectService.java

```
/**
 * 
 */
package projects.service;

import projects.dao.ProjectDao;
import projects.entity.Project;

/**
```

```
 * @author lixy4
 *
 */
public class ProjectService {
       private ProjectDao projectDao = new ProjectDao();

       public Project addProject(Project project) {
              return projectDao.insertProject(project);
       }
}
```

ProjectDao.java

```
/**
 *
 */
package projects.dao;

import java.math.BigDecimal;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

import projects.entity.Project;
import projects.exception.DbException;
import provided.util.DaoBase;

/**
 * @author lixy4
 *
 */
public class ProjectDao extends DaoBase {
       private static final String CATEGORY_TABLE = "category";
       private static final String MATERIAL_TABLE = "material";
       private static final String PROJECT_TABLE = "project";
       private static final String PROJECT_CATEGORY_TABLE = "project_category";
       private static final String STEP_TABLE = "step";

       public Project insertProject(Project project) {
              // @formatter:off
              String sql = ""
                     + "INSERT INTO " + PROJECT_TABLE + " "
                     + "(project_name, estimated_hours, actual_hours, difficulty,
notes)"
                     + "VALUES"
                     + "(?, ?, ?, ?, ?)";
              // @formatter:on
              try (Connection conn = DbConnection.getConnection()){
                     startTransaction(conn);

                     try(PreparedStatement stmt = conn.prepareStatement(sql)){
                            setParameter(stmt, 1, project.getProjectName(),
String.class);
```

```
                                setParameter(stmt, 2, project.getEstimatedHours(), 
BigDecimal.class);
                                setParameter(stmt, 3, project.getActualHours(), 
BigDecimal.class);
                                setParameter(stmt, 4, project.getDifficulty(), 
Integer.class);
                                setParameter(stmt, 5, project.getNotes(), 
String.class);

                                stmt.executeUpdate();

                                Integer projectId = getLastInsertId(conn, 
PROJECT_TABLE);
                                commitTransaction(conn);

                                project.setProjectId(projectId);
                                return project;
                        }
                        catch(Exception e) {
                                rollbackTransaction(conn);
                                throw new DbException(e);
                        }
                }
                catch(SQLException e) {
                        throw new DbException(e);
                }
        }
}
```