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

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

/**
 *
 *  @author jamie
 *
 */
public class Projects {
    private Scanner scanner = new Scanner(System.in);
    private ProjectService projectService = new ProjectService();
    private Project curProject;

    // @formatter:off
    private List<String> operations = List.of(
            "1) add a project",
            "2) List projects",
            "3) Select a project"
            );
    // @formatter:on


    /**
     * @param args
     */


    public static void main(String[] args) {
        new Projects().processUserSelections();
    }
    private void processUserSelections() {
        boolean done = false;

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

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

                case 1:
                    createProject();
                    break;

                case 2:
```

```
 58                      listProjects();
 59                      break;
 60
 61                  case 3:
 62                      selectProject();
 63                      break;
 64
 65                  default:
 66                      System.out.println("\n" + selection + " is not a valid selection. Try
    again.");
 67                      break;
 68                  }
 69              }
 70              catch(Exception e) {
 71                  System.out.println("\nError: " + e + " Try again.");
 72              }
 73
 74          }
 75
 76      }
 77      private void selectProject() {
 78          listProjects();
 79          Integer projectId = getIntInput("Enter a project ID to select a project");
 80
 81          curProject = null;
 82
 83          curProject = projectService.fetchProjectById(projectId);
 84
 85      }
 86      private void listProjects() {
 87          List<Project> projects = projectService.fetchAllProjects();
 88
 89          System.out.println("\nProjects:");
 90
 91          projects.forEach(project -> System.out.println("    " + project.getProjectId() + ": "
    + project.getProjectName()));
 92
 93      }
 94
 95      private void createProject() {
 96          String projectName = getStringInput("Enter the project name");
 97          BigDecimal estimatedHours = getBigDecimalInput("Enter the estimated hours");
 98          BigDecimal actualHours = getBigDecimalInput("Enter the actual hours");
 99          Integer difficulty = getIntInput("Enter the project difficulty (1-5)");
100          String notes = getStringInput("Enter the project notes");
101
102          Project project = new Project();
103
104          project.setProjectName(projectName);
105          project.setEstimatedHours(estimatedHours);
106          project.setActualHours(actualHours);
107          project.setDifficulty(difficulty);
108          project.setNotes(notes);
109
110          Project dbProject = projectService.addProject(project);
111          System.out.println("You have successfully created project: " + dbProject);
112      }
```

```
113
114     private BigDecimal getBigDecimalInput(String prompt) {
115         String input = getStringInput(prompt);
116
117         if(Objects.isNull(input)) {
118         return null;
119     }
120
121     try {
122         return new BigDecimal(input).setScale(2);
123     }
124     catch(NumberFormatException e) {
125     throw new DbException(input + " is not a valid decimal number.");
126 }
127     }
128
129 /*
130  * @return {@code true}
131  */
132 private boolean exitMenu() {
133     System.out.println("Exiting the menu.");
134     return true;
135 }
136     /*
137      * @return the menu selection as an int or -1 if nothing is selcted.
138      */
139     private int getUserSelection() {
140         printOperations();
141
142         Integer input = getIntInput("Enter a menu selection");
143
144         return Objects.isNull(input) ? -1 : input;
145     }
146
147     private Integer getIntInput(String prompt) {
148         String input = getStringInput(prompt);
149
150         if(Objects.isNull(input)) {
151             return null;
152         }
153
154         try {
155             return Integer.valueOf(input);
156         }
157         catch(NumberFormatException e) {
158             throw new DbException(input + " is not a valid number.");
159         }
160     }
161     private String getStringInput(String prompt ) {
162         System.out.println(prompt + ": ");
163         String input = scanner.nextLine();
164
165         return input.isBlank() ? null : input.trim();
166     }
167
168     private void printOperations() {
169         System.out.println("\nThese are the available selections. Press the Enter key to
```

```
   quit:");
170
171         operations.forEach(line -> System.out.println(" " + line));
172
173         if(Objects.isNull(curProject)) {
174             System.out.println("\nYou are not working with a project.");
175         } else {
176             System.out.println("\nYou are working with project: " + curProject);
177         }
178
179         }
180     }
181
```