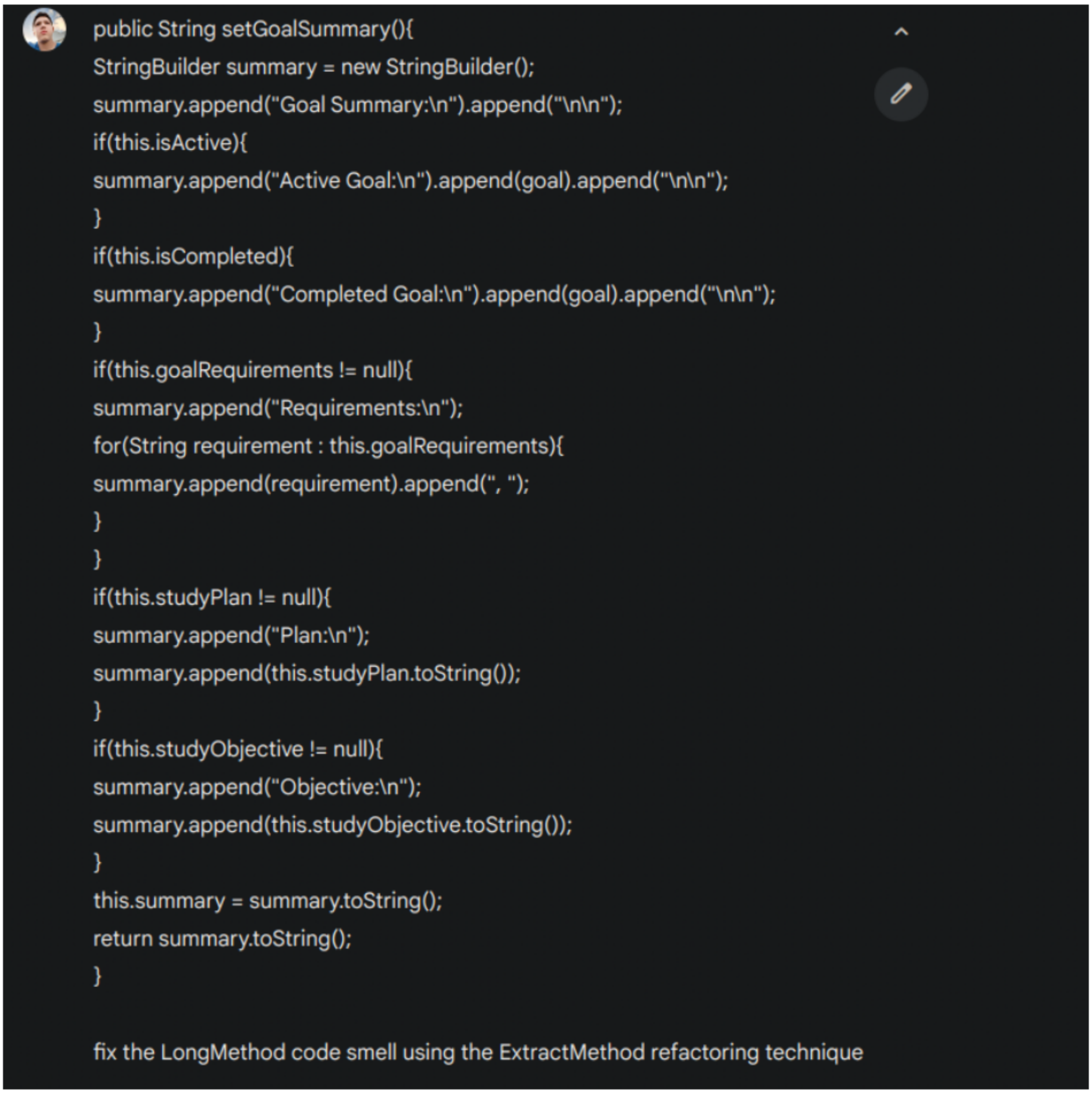
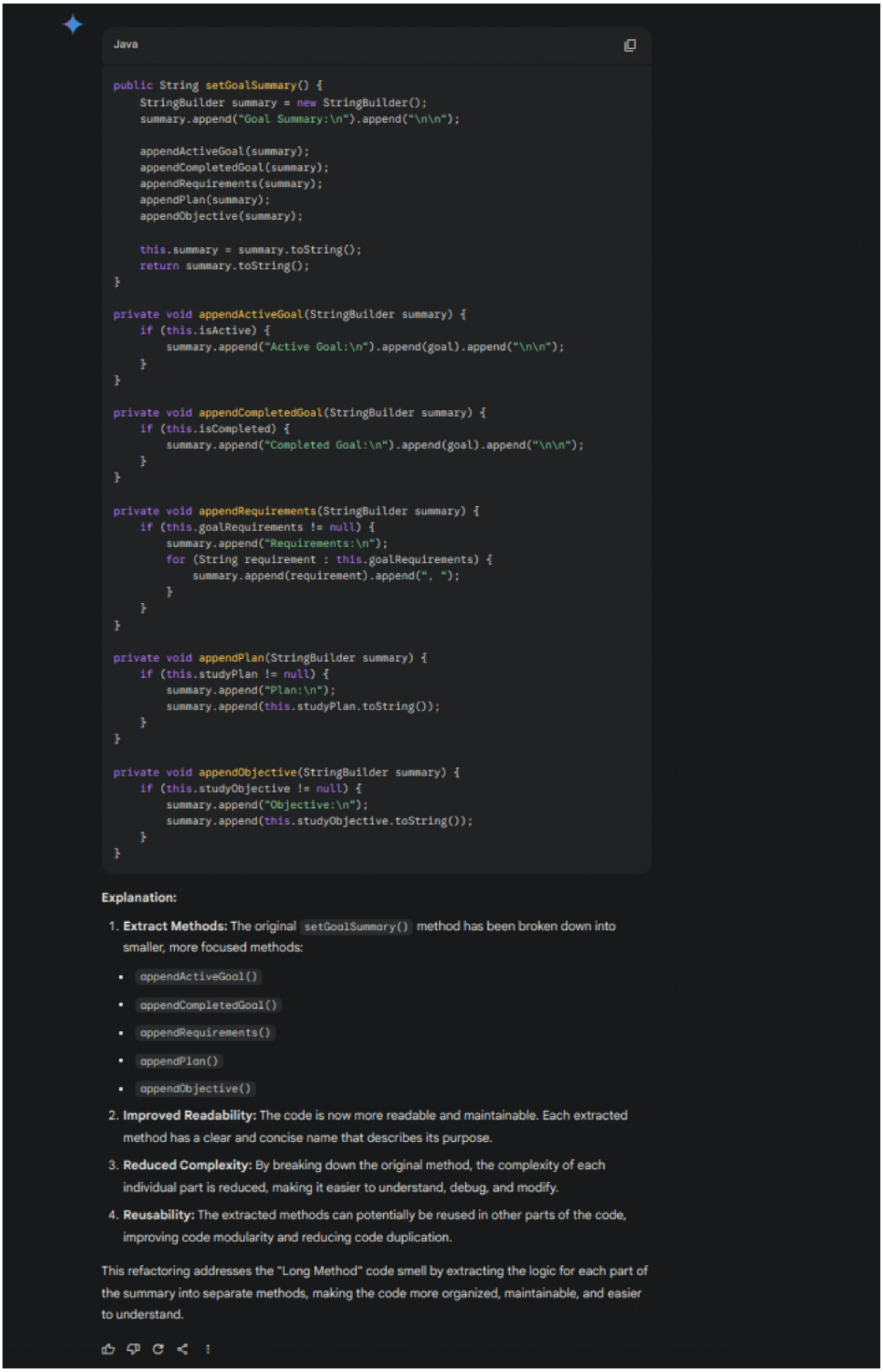
StudyGoal



```
public String setGoalSummary(){
StringBuilder summary = new StringBuilder();
summary.append("Goal Summary:\n").append("\n\n");
if(this.isActive){
summary.append("Active Goal:\n").append(goal).append("\n\n");
}
if(this.isCompleted){
summary.append("Completed Goal:\n").append(goal).append("\n\n");
}
if(this.goalRequirements != null){
summary.append("Requirements:\n");
for(String requirement : this.goalRequirements){
summary.append(requirement).append(", ");
}
}
if(this.studyPlan != null){
summary.append("Plan:\n");
summary.append(this.studyPlan.toString());
}
if(this.studyObjective != null){
summary.append("Objective:\n");
summary.append(this.studyObjective.toString());
}
this.summary = summary.toString();
return summary.toString();
}
```

fix the LongMethod code smell using the ExtractMethod refactoring technique



```java
public String setGoalSummary() {
    StringBuilder summary = new StringBuilder();
    summary.append("Goal Summary:\n").append("\n\n");

    appendActiveGoal(summary);
    appendCompletedGoal(summary);
    appendRequirements(summary);
    appendPlan(summary);
    appendObjective(summary);

    this.summary = summary.toString();
    return summary.toString();
}

private void appendActiveGoal(StringBuilder summary) {
    if (this.isActive) {
        summary.append("Active Goal:\n").append(goal).append("\n\n");
    }
}

private void appendCompletedGoal(StringBuilder summary) {
    if (this.isCompleted) {
        summary.append("Completed Goal:\n").append(goal).append("\n\n");
    }
}

private void appendRequirements(StringBuilder summary) {
    if (this.goalRequirements != null) {
        summary.append("Requirements:\n");
        for (String requirement : this.goalRequirements) {
            summary.append(requirement).append(", ");
        }
    }
}

private void appendPlan(StringBuilder summary) {
    if (this.studyPlan != null) {
        summary.append("Plan:\n");
        summary.append(this.studyPlan.toString());
    }
}

private void appendObjective(StringBuilder summary) {
    if (this.studyObjective != null) {
        summary.append("Objective:\n");
        summary.append(this.studyObjective.toString());
    }
}
```

Explanation:

1. **Extract Methods:** The original `setGoalSummary()` method has been broken down into smaller, more focused methods:
   - `appendActiveGoal()`
   - `appendCompletedGoal()`
   - `appendRequirements()`
   - `appendPlan()`
   - `appendObjective()`
2. **Improved Readability:** The code is now more readable and maintainable. Each extracted method has a clear and concise name that describes its purpose.
3. **Reduced Complexity:** By breaking down the original method, the complexity of each individual part is reduced, making it easier to understand, debug, and modify.
4. **Reusability:** The extracted methods can potentially be reused in other parts of the code, improving code modularity and reducing code duplication.

This refactoring addresses the "Long Method" code smell by extracting the logic for each part of the summary into separate methods, making the code more organized, maintainable, and easier to understand.