CS6.401 Software Engineering Spring 2024

Project- 1 Report (Bonus Task)

Team 23

Team Members:

Name	Roll Number
Karan Bhatt	2022202003
Madhusree Bera	2022202007
Vedashree Ranade	2022201073
Yash Maheshwari	2022201074
Piyush Rana	2022202012

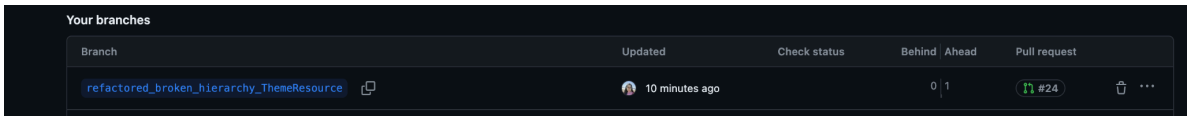
Bonus Task: Automated Refactoring Pipeline

Note:

- The task is implemented for a specific file instead of the entire repository considering the limitations of API usage (for GPT-3.50) imposed by openAI.
- **Bonus task Python file path**: [branch] bonus –> bonus_task –> bonus.ipynb

Procedure

- **Step 1: New branch creation in github**



- **Step 2: Automated Design Smell Detection and Refactoring**
  - Using the OpenAI API, detected the design smells by giving specific prompts to the LLM.
  - The code was refactored by utilizing the LLM, by providing prompts to do so.
  - Input File content provided to LLM:

```java
@Path("/theme")
public class ThemeResource extends BaseResource {
    /**
     * Returns the list of all themes.
     *
     * @return Response
     * @throws JSONException
     */
    @GET
    @Produces(MediaType.APPLICATION_JSON)
    public Response list() throws JSONException {
        ThemeDao themeDao = new ThemeDao();
        List<String> themeList = themeDao.findAll();
        JSONObject response = new JSONObject();
        List<JSONObject> items = new ArrayList<>();
        for (String theme : themeList) {
            JSONObject item = new JSONObject();
            item.put("id", theme);
            items.add(item);
        }
        response.put("themes", items);
        return Response.ok().entity(response).build();
    }
}
```

- Output given by LLM:

```
In [5]: 1 t1 = detect_broken_hierarchy(code_sample)

In [6]: 1 print(t1)
```

```
ChatCompletion(id='chatcmpl-8vRJtQ8jKVOfCXwRjDbkukNVf8a9n', choices=[Choice(finish_reason='stop', index=0, logprobs=
None, message=ChatCompletionMessage(content='In the given code, the ThemeResource class extends BaseResource but doe
s not use any of its methods. This indicates a broken hierarchy smell.\n\nTo refactor the code and remove the broken
hierarchy smell, we can remove the inheritance from BaseResource if it is not needed. If there are common methods th
at need to be shared between ThemeResource and other resources, those methods can be extracted into a separate class
and used as a composition instead of inheritance.\n\nRefactored code:\n```java\npackage com.sismics.books.rest.resou
rce;\n\nimport com.sismics.books.core.dao.file.theme.ThemeDao;\nimport org.codehaus.jettison.json.JSONException;\nim
port org.codehaus.jettison.json.JSONObject;\n\nimport javax.ws.rs.GET;\nimport javax.ws.rs.Path;\nimport javax.ws.r
s.Produces;\nimport javax.ws.rs.core.MediaType;\nimport javax.ws.rs.core.Response;\nimport java.util.ArrayList;\nimp
ort java.util.List;\n\n/**\n * Theme REST resources.\n * \n * @author jtremeaux\n */\n@Path("/theme")\npublic class
ThemeResource {\n    /**\n     * Returns the list of all themes.\n     * \n     * @return Response\n     * @throws J
SONException\n     */\n    @GET\n    @Produces(MediaType.APPLICATION_JSON)\n    public Response list() throws JSONEx
ception {\n        ThemeDao themeDao = new ThemeDao();\n        List<String> themeList = themeDao.findAll();\n
JSONObject response = new JSONObject();\n        List<JSONObject> items = new ArrayList<>();\n        for (String th
eme : themeList) {\n            JSONObject item = new JSONObject();\n            item.put("id", theme);\n
items.add(item);\n        }\n        response.put("themes", items);\n        return Response.ok().entity(response).b
uild();\n    }\n}\n```', role='assistant', function_call=None, tool_calls=None))], created=1708700405, model='gpt-3.
5-turbo-0125', object='chat.completion', system_fingerprint='fp_cbdb91ce3f', usage=CompletionUsage(completion_tokens
=343, prompt_tokens=312, total_tokens=655))
```

- **Step 3: Updating the file contents with the refactored code in the new branch**

- **Step 4: Created pull request to merge the refactored code with the branch “bonus”**

Tools and Technologies Used

- **GitHub API:** For accessing and monitoring GitHub repositories.
- **Language Models:** Utilize LLMs for design smell detection and refactoring.
- **Python:** Use Python for scripting and interacting with APIs.

Summary

The automated pipeline scans the code file, utilizes LLMs for design smell detection and refactoring, and generates pull requests.