Tanner Nantabutr, Asif Chowdury, David Nguyen, Sahaj Singh

Process Deliverable: Agile

- What went well: In our last sprint, our redesign and iteration of our product went well and we made good progress on taking the feedback we received and applying it to our design.
- What did not go as planned: At the beginning, we found it a little difficult to take into account all the ideas and iterations we had in mind. The task had seemed a little daunting but we managed to pick up the pace and finish the sprint with good progress.
- What can we do better: For our next sprint, we will ensure to create a more cohesive plan and work out kinks and details before jumping into the task at hand.
- Prioritized tasks (sprint planning): Definitive designing using high and low-level design. Create sketches and visualize product usage among users.

High Level Design

I believe that for our application the Builder Design Pattern is the best choice. This pattern is particularly well-suited for applications that involve creating complex objects step-by-step, making it ideal for our application.

For our application, the design can be divided into distinct components. One component would be responsible for the overall process of extracting, organizing, and presenting TODO comments. This component would give specific tasks to the Builder, which encapsulates the logic for each step, such as file parsing, comment extraction, and output formatting. The last component would implement these steps for different formats. All of these components work together to create the working application.

Using the Builder Design Pattern provides several advantages. It introduces modularity by separating the construction process into manageable steps, making it easier to modify the functionality of the application. For example, adding support for a new output format or programming language would only require creating a new builder without altering the core logic. This modularity also improves maintainability, as changes to one component do not affect the others, reducing the risk of errors.

Low Level Design

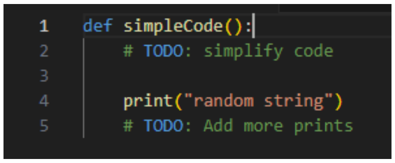
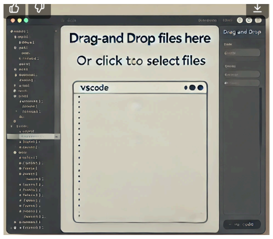
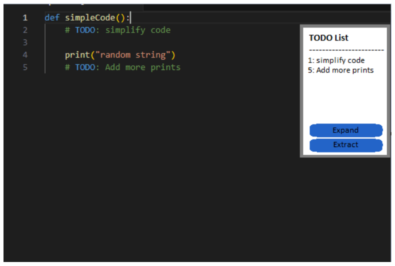
To implement the extraction and compilation of TODO comments, the Builder Design Pattern is ideal. It enables separating the logic for extracting TODOs from files and formatting the output into a structured list (e.g., text or JSON). This modular approach makes the system easier to maintain and extend for new formats or languages.

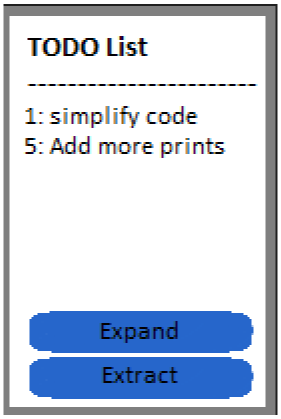
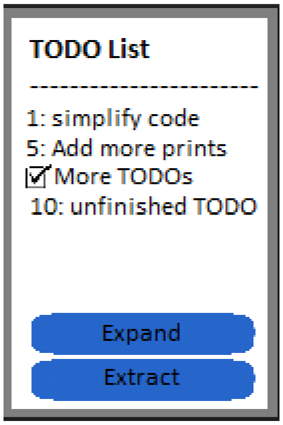
Example Pseudocode:

```
class TodoListBuilder:
    def __init__(self):
        self.todos = []

    def extract(self, file):
        for i, line in enumerate(open(file), 1):
            if "TODO" in line:
                self.todos.append(f"{file}:{i} - {line.strip()}")

    def build(self):
        return "\n".join(self.todos)

builder = TodoListBuilder()
for file in ["file1.py", "file2.js"]:
    builder.extract(file)
print(builder.build())
```

Design Sketch

Simplify Your TODOs	Select Files, Get TODOs	Your TODOs, Organized


def simpleCode():
    # TODO: simplify code
 	
    print("random string")
    # TODO: Add more prints
Discover the easiest way to extract and organize TODOs from your codebase! Click "Start Now" to begin.
	
	Drag-and Drop files here
	Or click tco select files
	
	Easily drag and drop files or choose from your system to locate TODO comments in seconds.
		
		def simpleCode():
		    # TODO: simplify code
		
		    print("random string")
		    # TODO: Add more prints
		TODO List
		----------------------- 	
		1: simplify code
		5: Add more prints
		
		Save time and reduce stress by having all your TODO comments structured and ready for action. No more manual searching!

Extract TODOs Now	Quick and Easy Setup	Keep It Organized
 TODO List ------------------------ 1: simplify code 5: Add more prints Simply click "Extract TODOs" and watch as the app processes your files to generate a clear, actionable list.	 TODO List ------------------------ 1: simplify code 5: Add more prints ☑ More TODOs 10: unfinished TODO 1. Upload your files. 2. Click "Extract TODOs." 3. Review and save your TODOs in seconds.	 EXPORT TODO LIST EXPORT EXPORT JSON JSON LIST EXPORT TEXT FILES Export TODOs as JSON or text files for seamless project management. Revisit the app anytime to stay on track.