

Phase 2 Presentation



Group Members: Aliyan, Bernard, Ruiting, Tiffany, Miguel, Linh, Anita, Leya

Updated Specification

Domain: Financial Planning



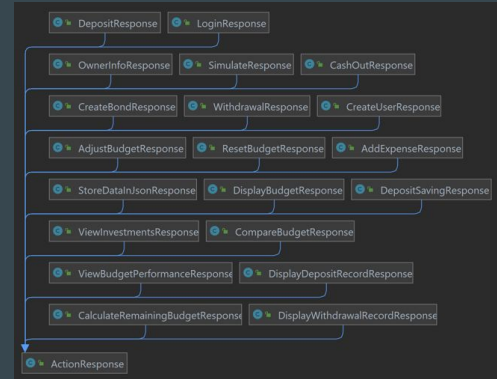
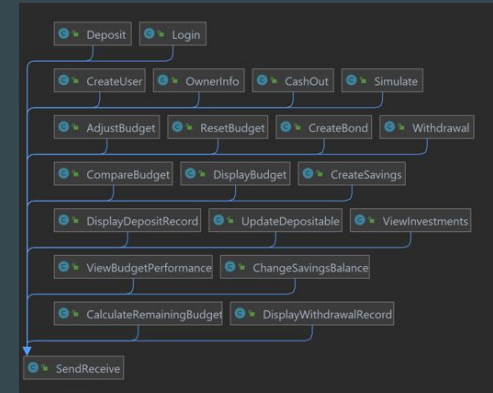
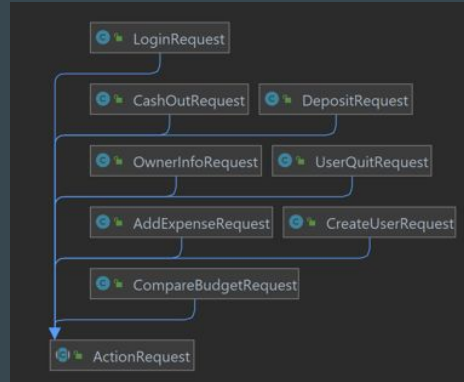
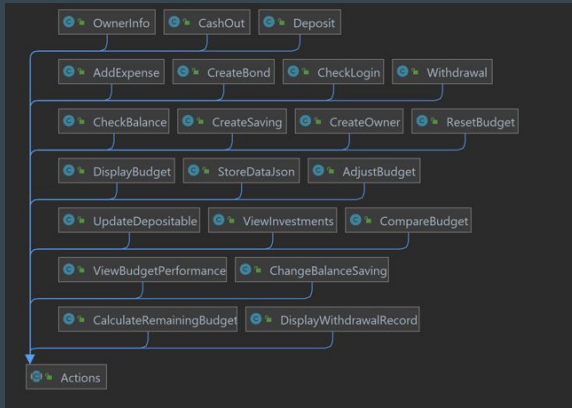
Clean Architecture and Scenario Walkthrough

SOLID Principles

- Single responsibility Principle
- Open-Closed Principle
- Liskov substitution principle
- Interface Segregation principle
- Dependency inversion principle

Class Diagram of Features

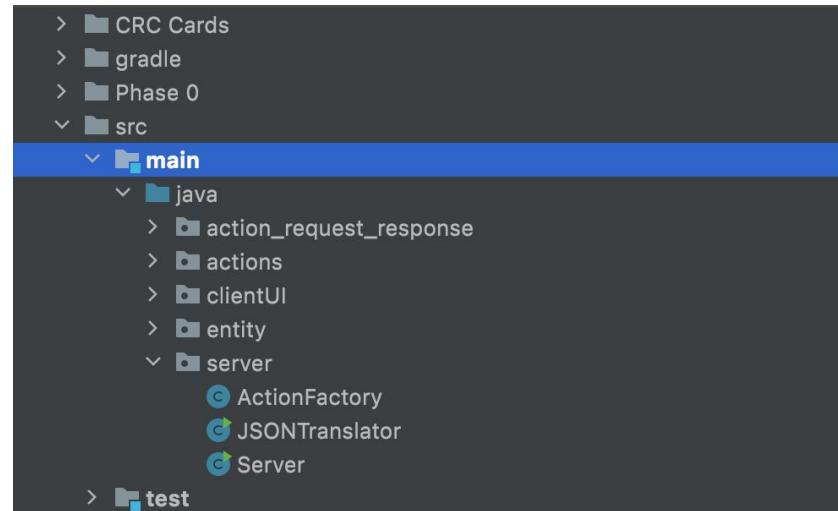
Use Case Diagram



Accessibility Report

Major Design Decisions

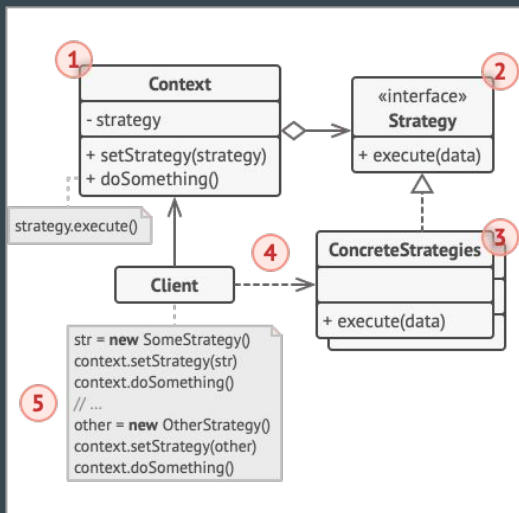
Packaging Strategies



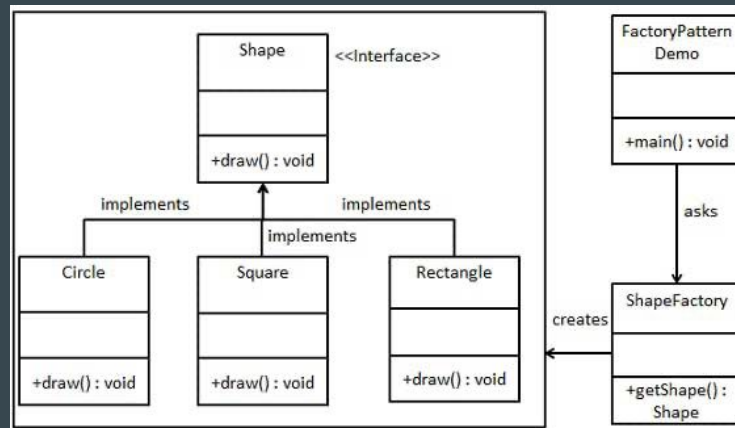
- Used the packaging by layer strategy
-

Design Pattern

- Strategy Design Pattern (Pull request #100):
 - Used in the Simulare Class in the Action package.



- Factory Design Pattern (Pull request #29):
 - Used in ActionFactory in Server package



Refactor

Github Features

- Utilized the Issues and Review/Conversation tab to identify any potential issues.

The screenshot shows the GitHub Issues page for the 'a-gohar' repository. The top navigation bar includes links for Code, Issues (2), Pull requests, Actions, Projects (1), Wiki, Security, Insights, and Settings. A message states: 'Label issues and pull requests for new contributors. Now, GitHub will help potential first-time contributors discover issues labeled with **good first issue**.' Below this is a search bar with the query 'is:issue is:closed' and buttons for 'Labels 9', 'Milestones 0', and 'New issue'. A list of issues is displayed, all marked as 'Closed':

Issue Title	Author	Label	Projects	Milestones	Assignee	Sort
Tests failing because they weren't updated to the new actionfactory design	#93 by a-gohar	was closed 3 days ago				1
BudgetHistory violates clean architecture	#89 by by20c	was closed 4 days ago				2
Date does not take into account leap years and leap days.	#88 by a-gohar	was closed 5 days ago				1
Server doesn't appear to close a connection if incorrect object sent from ClientUserInterface	#21 by a-gohar	was closed on Oct 31				1

The screenshot shows the 'Conversation' tab of a GitHub Pull Request (PR) #1140b98. The PR title is 'Serialization Works, we need to add implements serialization to class...'. The conversation includes:

- A comment from **MJAquarion2413** (Member) 5 days ago: 'Will work but needs classes that will be serializable, will change that once I'm absolutely sure that serialization wont have any adverse effects. Testing area at bottom of code with example'.
- A review from **ruiting-chen** (Member) 5 days ago: 'I would suggest renaming this class to ReadWriter instead of Serialization, so that it is more clear what this class does. And rename the methods Serialize and Deserialize to saveToFile and readFromFile, so that it is expressed what the methods do more explicitly.'
- A review from **leya-595** (Member) 4 days ago: 'Add a bit more documentation for serialize and deserialize to explain the methods purpose.'

The code snippet shown in the PR is:

```
src/main/java/server/Serialization.java
22 + }
23 + }
24 +
25 + public void Serialize(Object obj) throws IOException {
```

What Worked Well

- Factory Design allowed implementation of new actions to be straightforward.
- The workflow of the program allows for easy extensions of usable tools and implementation of completely new tools.

