



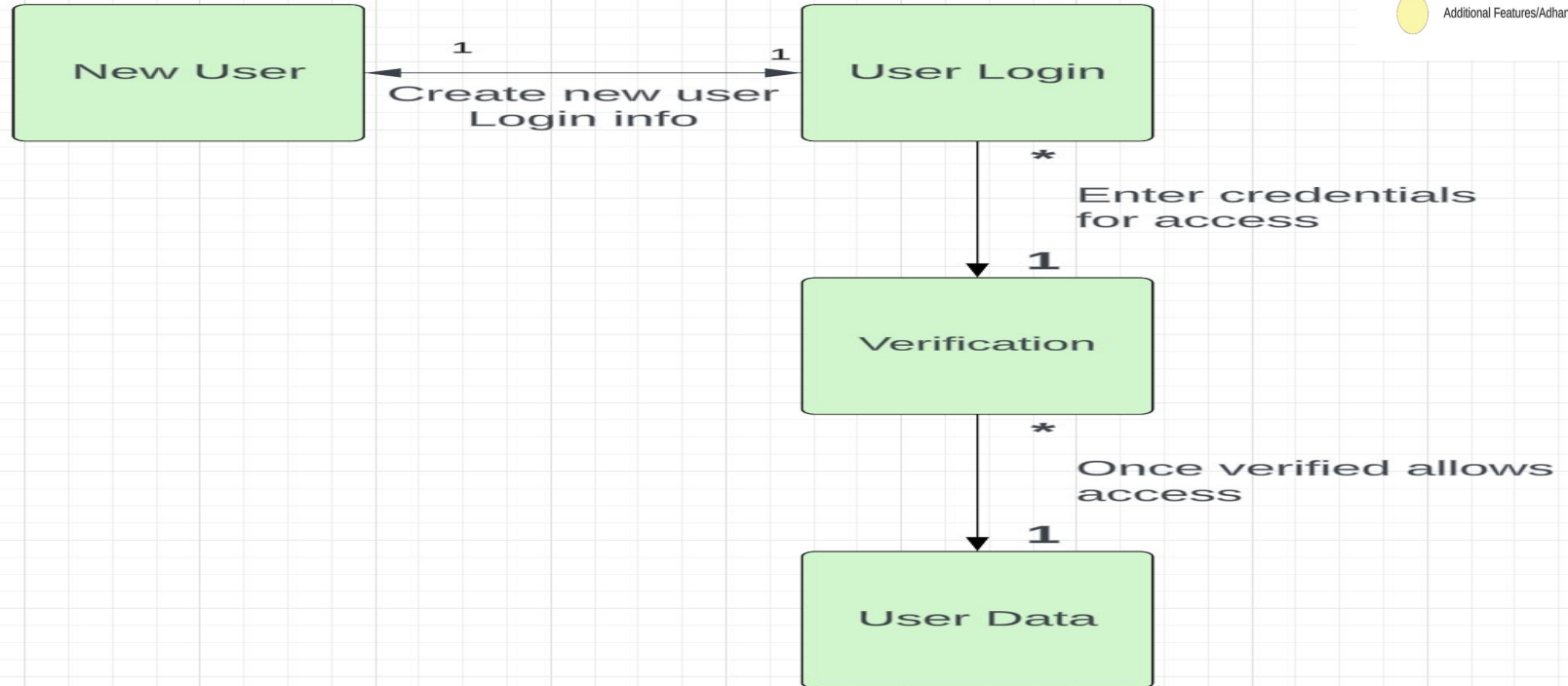
# Epic-Maven U-Fund Presentation.

By: Alex, Daniel, Kevin, Muhammad, and Red.

# Domain Model #1.

Index

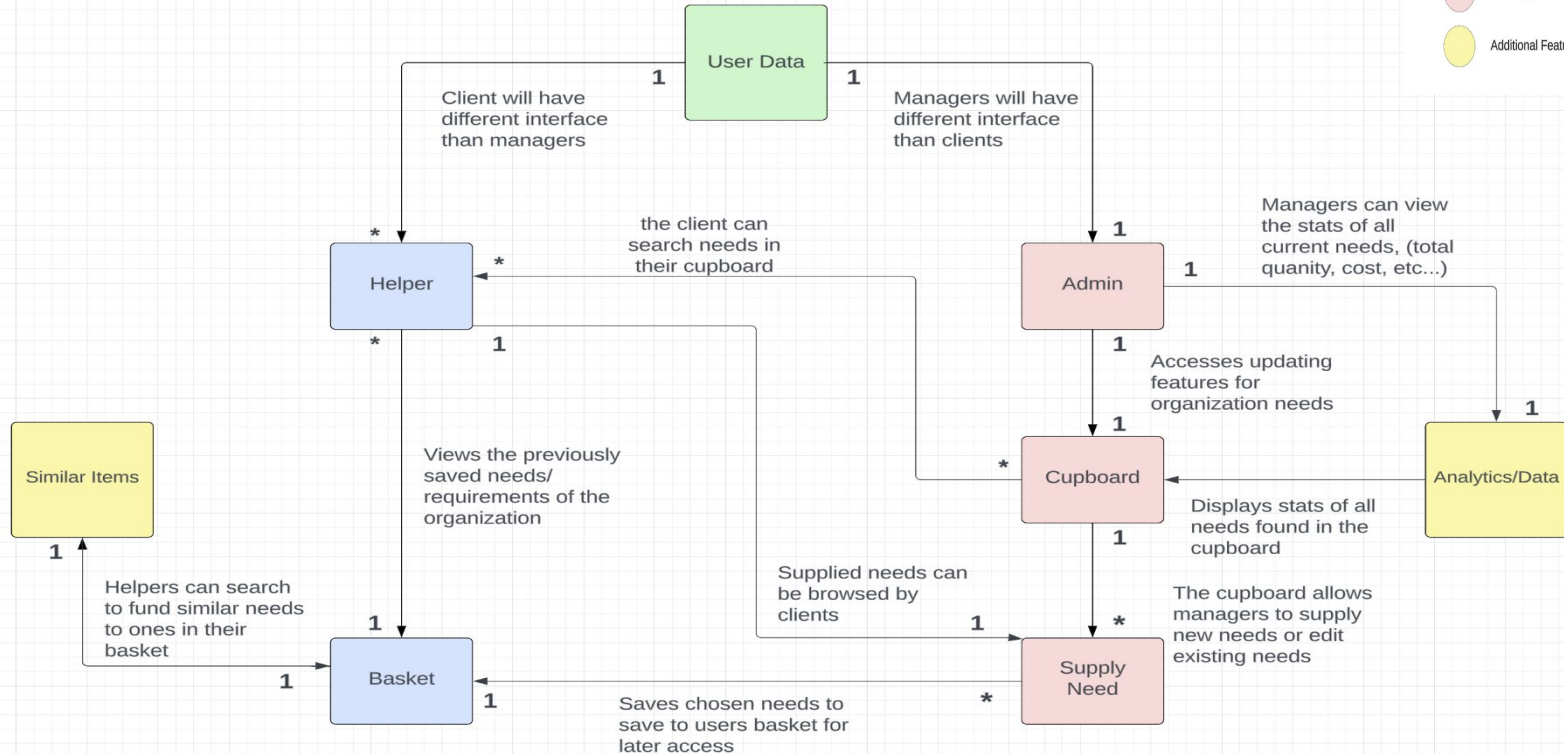
- Login/Verification (All Users)
- User/Helper model
- Manager/Admin model
- Additional Features/Adhancements



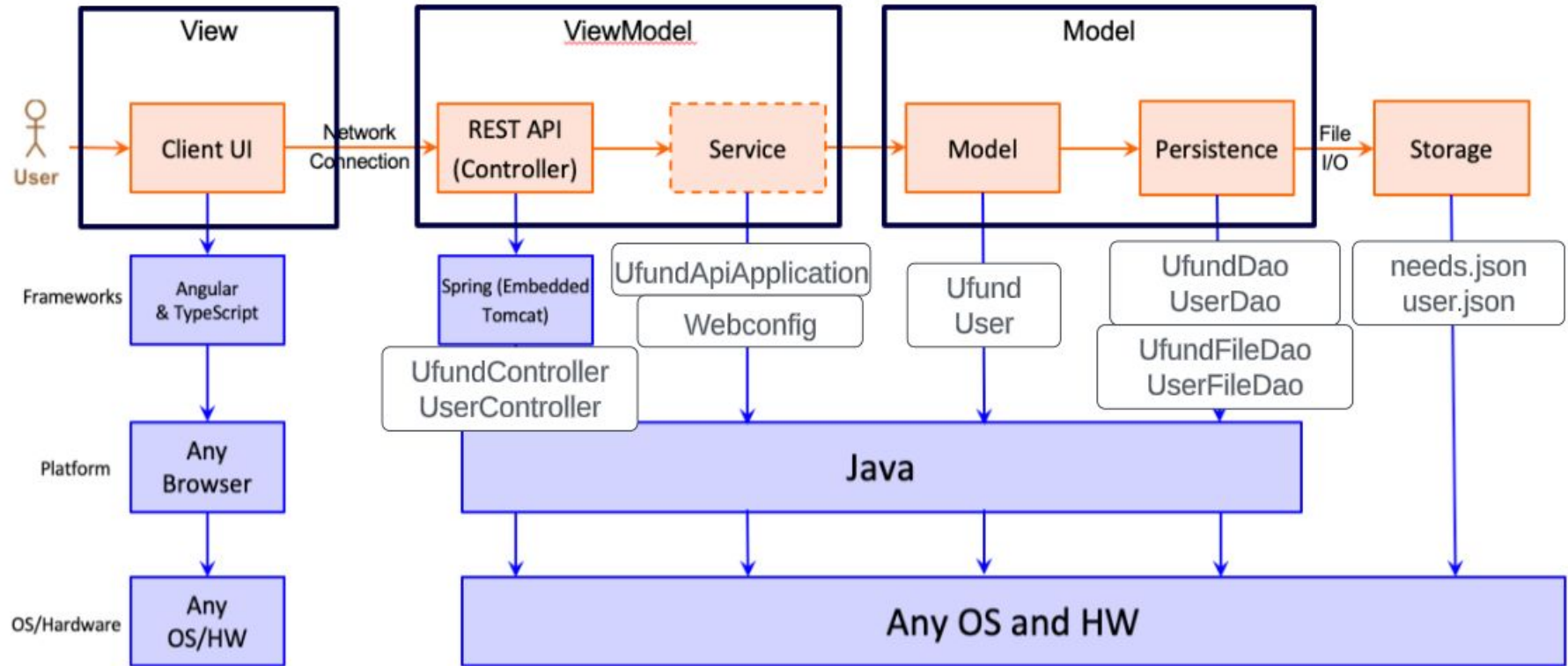
# Domain Model #2.

## Index

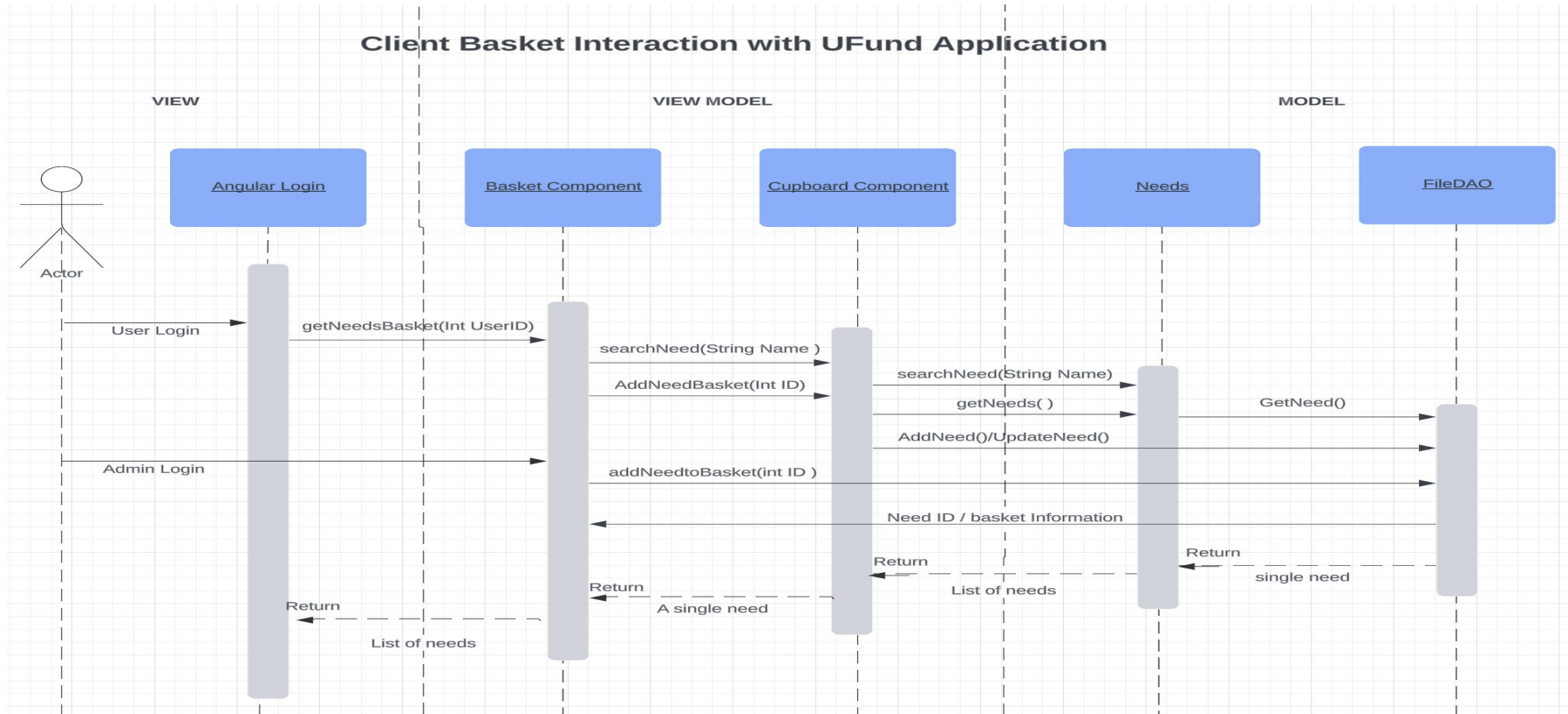
- Login/Verification (All Users)
- User/Helper model
- Manager/Admin model
- Additional Features/Advancements



# Architectural Model.

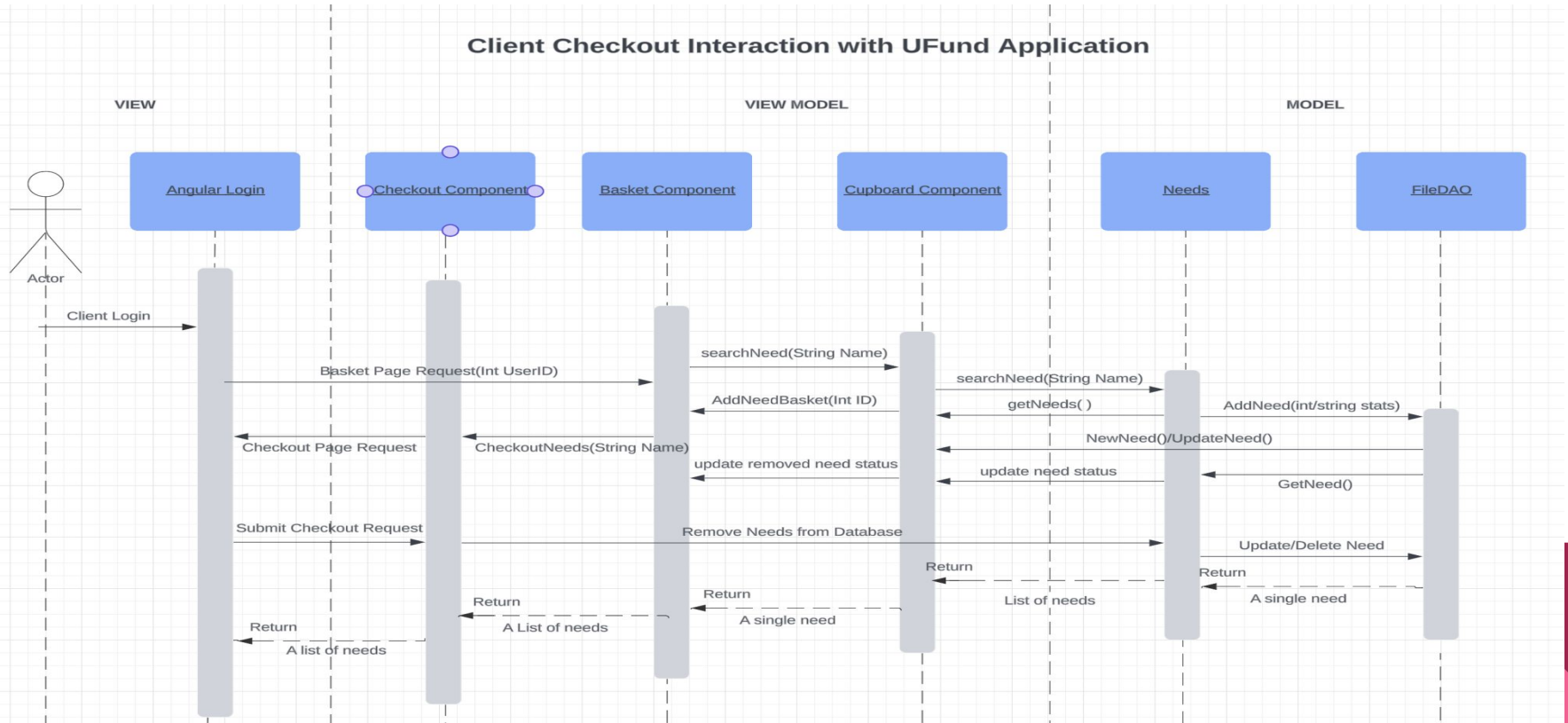


# Sequence Diagram #1.



# Sequence Diagram #2.

Client Checkout Interaction with UFund Application



# Static Code Analysis #1.

```
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 47, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jacoco:0.8.7:report (report) @ ufund-api ---
[INFO] Loading execution data file C:\Users\yousof\Sven 261 - Spring\team-project-rit-s
wen-261-02-5e-epicmaven\ufund-api\target\jacoco.exec
[INFO] Analyzed bundle 'ufund-api' with 8 classes
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 15.789 s
[INFO] Finished at: 2024-04-12T17:46:55-04:00
[INFO] -----

yousof@Yousof MINGW64 ~/Sven 261 - Spring/team-project-rit-swen-261-02-5e-epicmaven/ufu
nd-api (main)
$
```

The screenshot displays the SonarQube web interface for the 'ufund-api' project. The 'Measures' tab is active, showing a table of code coverage metrics for various Java files. The overall coverage is 77.1%.

File	Coverage	Uncovered Lines	Uncovered Conditions
src/main/java/com/ufund/api/ufundapi/Application.java	33.3%	2	33
src/main/java/com/ufund/api/ufundapi/persistence/ufundapi/dao/dao.java	63.8%	24	5
src/main/java/com/ufund/api/ufundapi/persistence/ufundapi/dao/dao.java	64.1%	22	6
src/main/java/com/ufund/api/ufundapi/model/ufundapi.java	66.7%	6	67
src/main/java/com/ufund/api/ufundapi/model/ufundapi.java	71.4%	6	79
src/main/java/com/ufund/api/ufundapi/controller/ufundapi/ufundapi.java	100%	0	0
src/main/java/com/ufund/api/ufundapi/controller/ufundapi/ufundapi.java	100%	0	0
src/main/java/com/ufund/api/ufundapi/WebConfig.java	100%	0	100

8 of 8 shown

# Static Code Analysis #2.

The image displays two side-by-side screenshots of the SonarQube web interface, showing the results of a static code analysis for a project named 'ufund-app'.

**Left Screenshot (Main View):**

- Header:** 'main' branch, 1.4k Lines of Code, Version 0.0.1-ONAPSPOT.
- Quality Gate Status:** **Passed** (Green checkmark). A message says: "Enjoy your sparkling clean code!"
- Measures:** Last analysis 2 hours ago.
  - Security:** 0 Open Issues (A).
  - Reliability:** 0 Open Issues (A).
  - Maintainability:** 94 Open Issues (A).
  - Coverage:** 77.1% (G). 200 New Lines to cover.
  - Duplications:** 0.0% (G). 1.4k New Lines.
  - Security Hotspots:** 0 (A).
- ACTIVITY:** There isn't enough data to generate an activity graph.
- Footer:** April 12, 2024 at 3:18 PM. Quality Gate: **Passed**. First analysis: 0 Issues - 77.1% Coverage - 0.0% Duplications.

**Right Screenshot (Measures View):**

- Header:** 'main' branch, 5.4k Lines of Code, Version not provided.
- Quality Gate Status:** **Passed** (Green checkmark). A message says: "Enjoy your sparkling clean code!"
- Measures:** Last analysis 3 hours ago.
  - Security:** 0 Open Issues (A).
  - Reliability:** 9 Open Issues (D).
  - Maintainability:** 54 Open Issues (A).
  - Coverage:** 0.0% (R). 69% New Lines to cover.
  - Duplications:** 15.7% (G). 5.4k New Lines.
  - Security Hotspots:** 4 (E).
- ACTIVITY:** There isn't enough data to generate an activity graph.
- Footer:** April 12, 2024 at 2:51 PM. Quality Gate: **Passed**. First analysis: 0 Issues - 0.0% Coverage - 15.7% Duplications.



# Recommendations.

- By establishing feedback to gather input from users, the system will be able to continuously improve over time.
- By incorporating security measures into the system like authentication, authorization, encryption, and data validation, there will be less impact from threats and vulnerabilities.
- By monitoring the system to track health, performance, and errors, there will be more solutions and insights on the system's behavior.



# Improvements #1.

- Dynamic content, personalized dashboards, and adaptive interfaces that adjust based on user history.
- Separate the cupboard, archive, and create need features. Wouldn't need to scroll past cupboard and archive to create a need.
- Switch exit button to top. Don't need to scroll past anything to go back.

## Cupboard:

Name: hot dog, Cost: \$1.00, Quantity: 100, Type: food

Name: shoes, Cost: \$2.00, Quantity: 5, Type: clothing

Name: cap, Cost: \$2.50, Quantity: 10, Type: clothing

Name: car, Cost: \$10000.00, Quantity: 8, Type: vehicle

Name: phone, Cost: \$1000.00, Quantity: 5, Type: electronic

Name: laptop, Cost: \$1000.00, Quantity: 10, Type: electronic

## Archive.

Name: hot soup, Cost: \$1.50, Quantity: 0, Type: food

Name: bus, Cost: \$100000.00, Quantity: 0, Type: vehicle

## Create

Name:

Cost:

Quantity:

Type:

Name: hot dog, Cost: \$1.00, Quantity: 100, Type: food

## Improvements #2.

- When editing needs, have the old values appear as a placeholder. More intuitive to realize you don't have to retype values you don't want to change.

Name.
Cost.
Quantity.
Type.
<input type="button" value="Update"/>

Name: hot dog, Cost: \$1.00, Quantity: 100, Type: food

hot dog
1
100
food
<input type="button" value="Update"/>



# Thank You.

By: Alex, Daniel, Kevin, Muhammad, and Red.