

HEART STRINGS

FINAL
PRESENTATION





01

Our Mission

A mission statement pulling at your heart strings!

03

Architectural Overview

- + In Other Words...
- + Improvement Recommendations

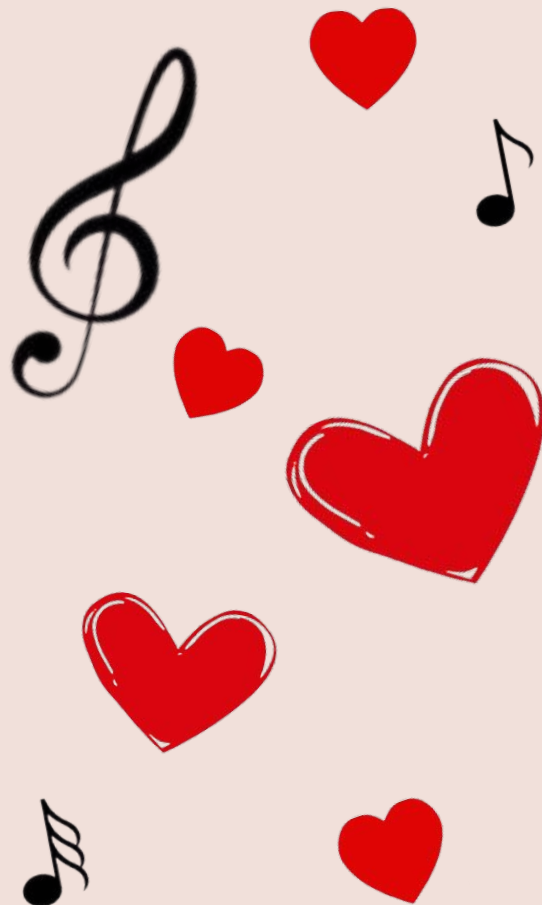


02

Final Domain Analysis

04

Sequence Diagram





05

Conception of Inventory

- + Domain Model Inventory System
- + Conception of the system

07

Helper Funding Basket UML



06

Similar Design Structures

08

Domain Model (Concert Enhancement)





09

Static Code Analysis Highlights



10

Usability Improvements

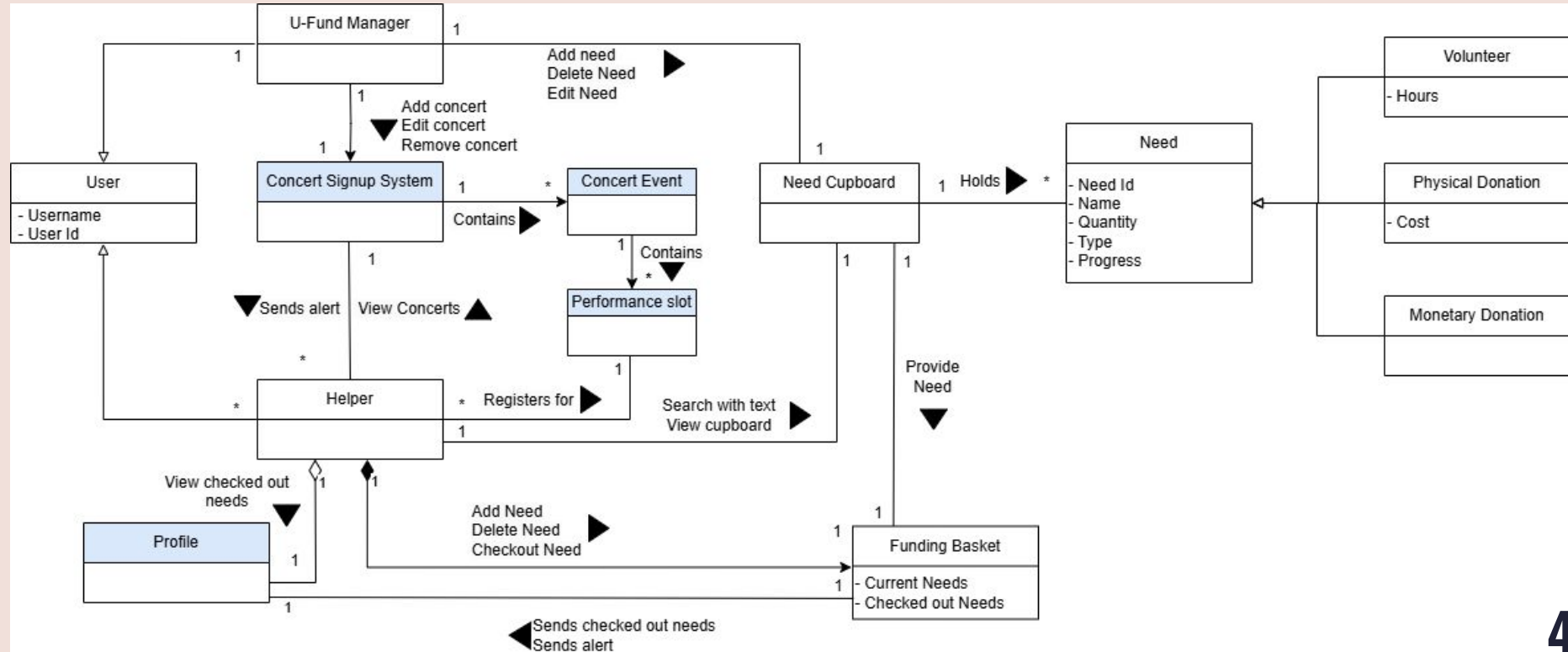




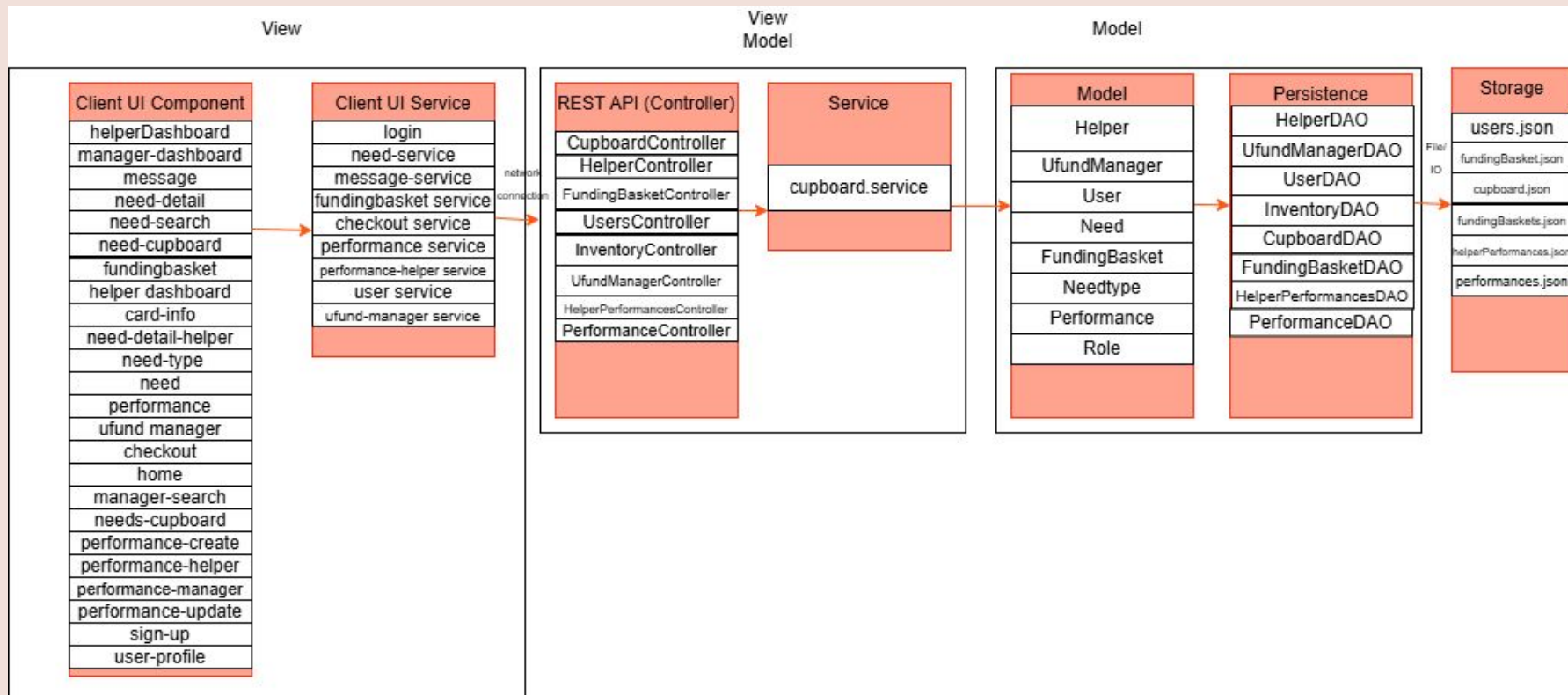
OUR MISSION STATEMENT

Our mission is to bring music to everyone who wants to participate, no matter the circumstances. Every helper who decides to donate their time, old instruments, or money helps us to give opportunities to a future musician and long lasting bonds to brilliant people.

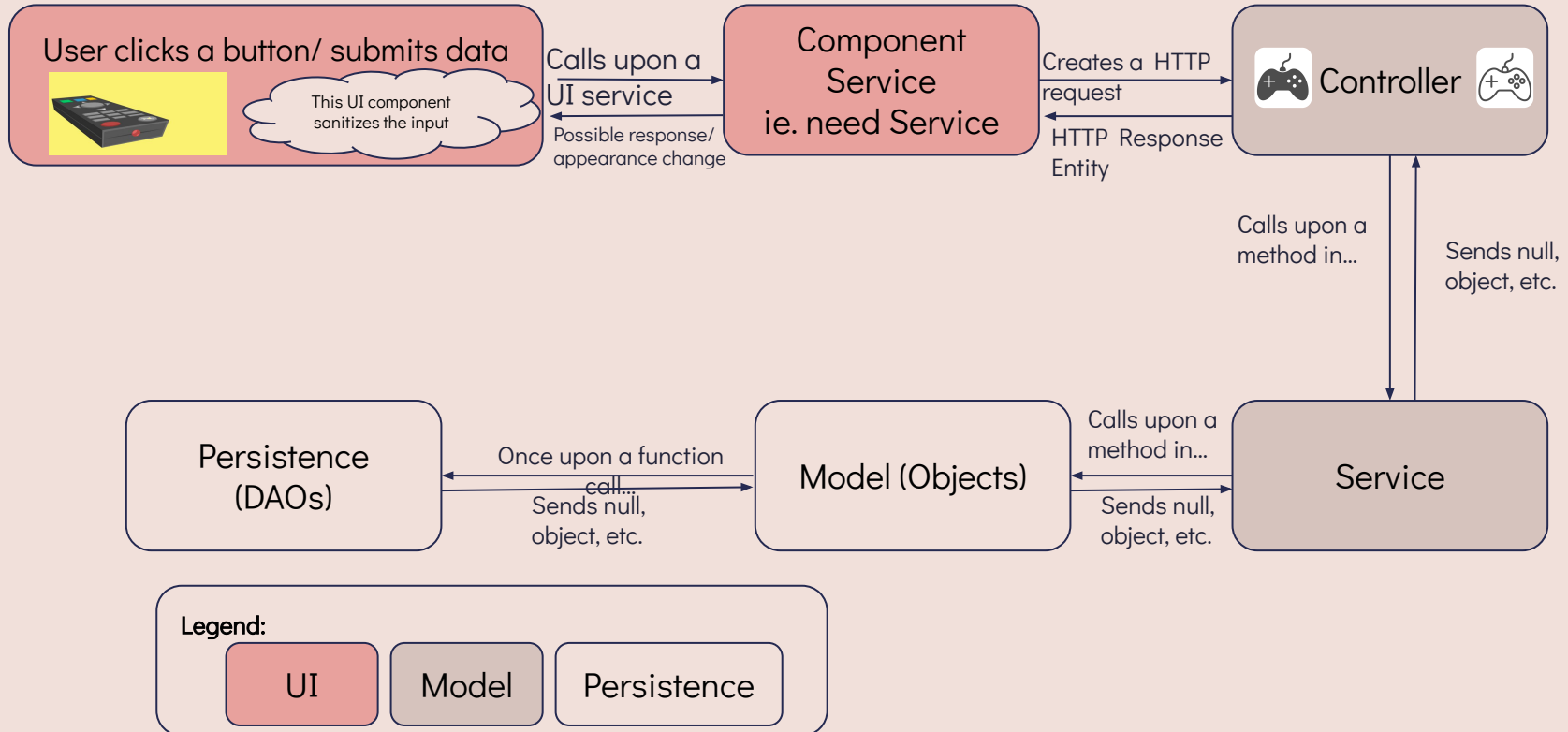
Final Domain Analysis



Architecture Overview



In other words...



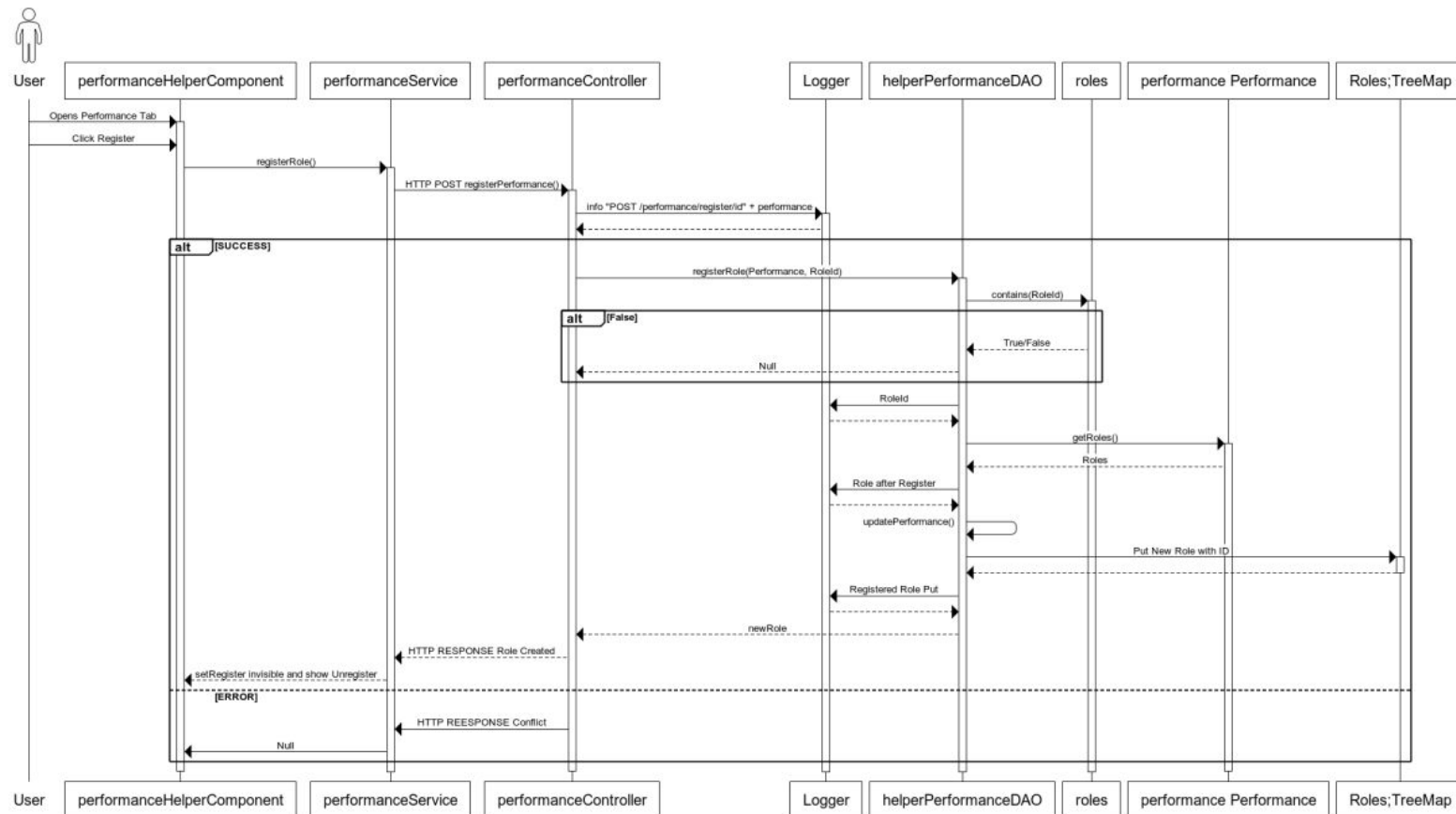
Software Architecture

Design Improvement Recommendations



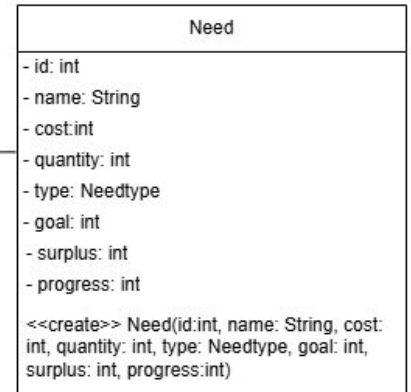
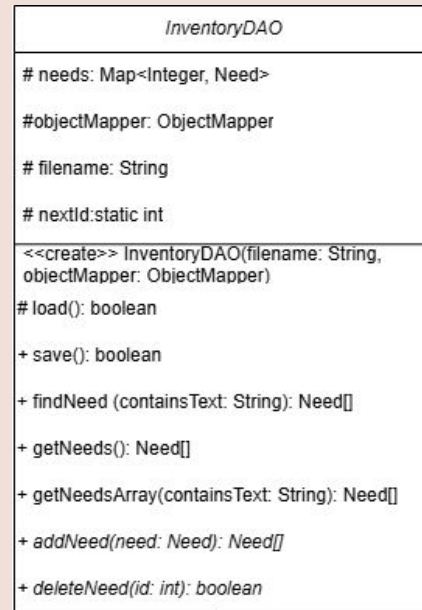
- ★ Funding Basket should have its own backend service, akin to Cupboard's.
- ★ Function duplication should be reduced and removed where possible, especially when the duplicated functions are frivolous and unused.
- ★ Remove all polling done as it is inefficient and can be done easier.

+ Sequence Diagram



+ Domain Model Inventory System

- Abstract class that defines similar functions i.e. loading, saving, and finding a need.
- The abstract methods were then meant to represent the different ways the DAOs handle similar functions
- This helps us maintain single responsibility in reference to the cupboard DAO and fundingBasketDAO
- Could allow us to easily switch out cupboard DAO in the future



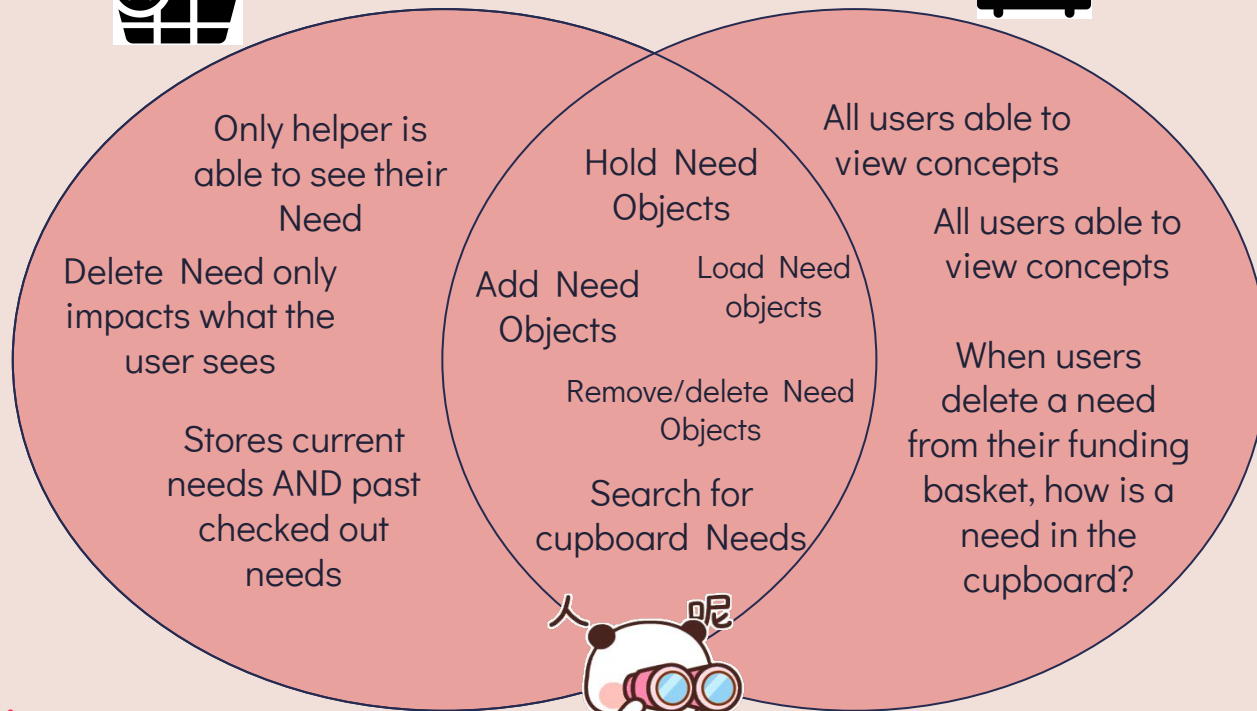
Conception of Inventory



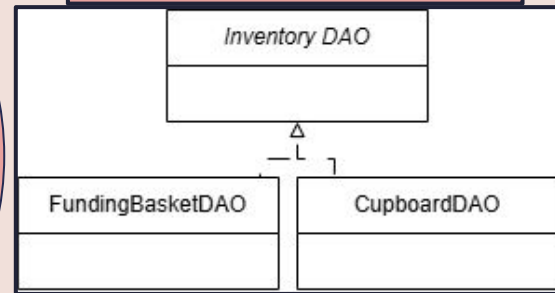
Funding Basket



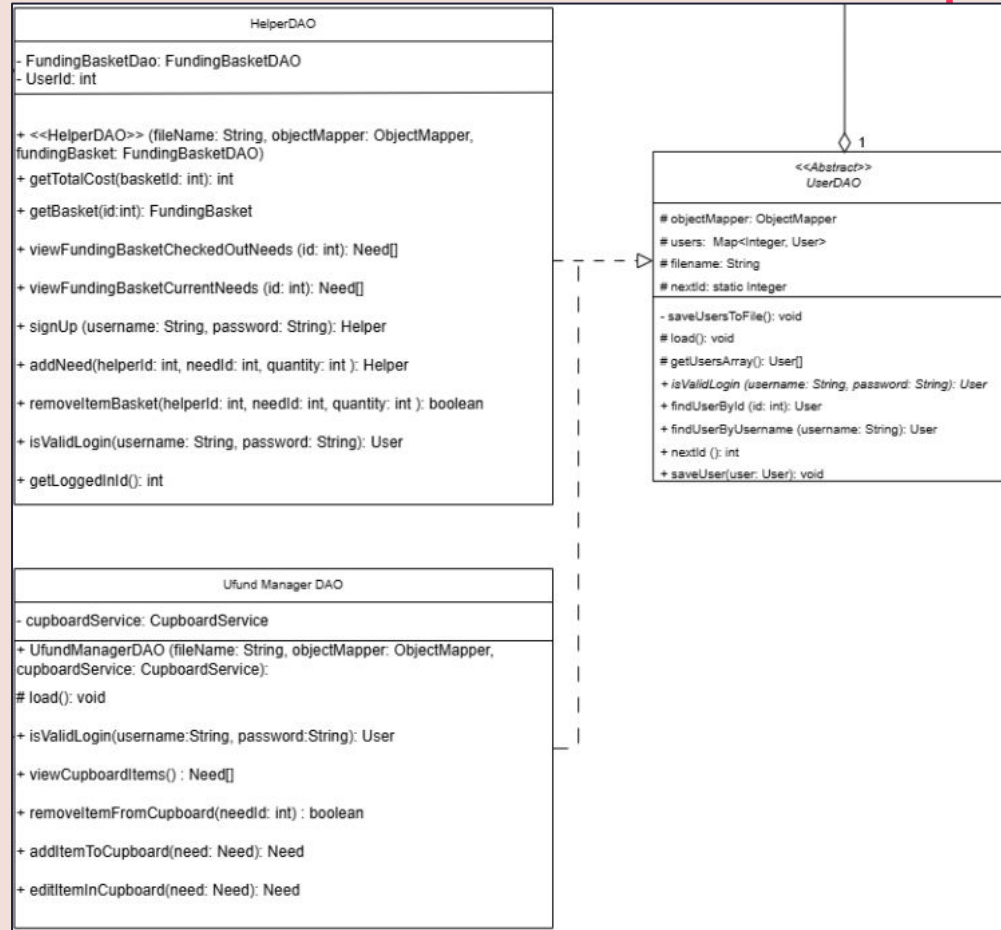
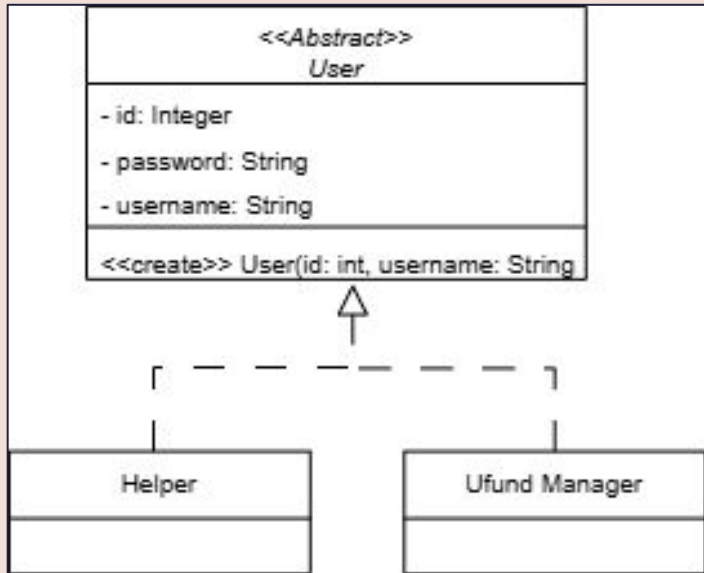
Cupboard



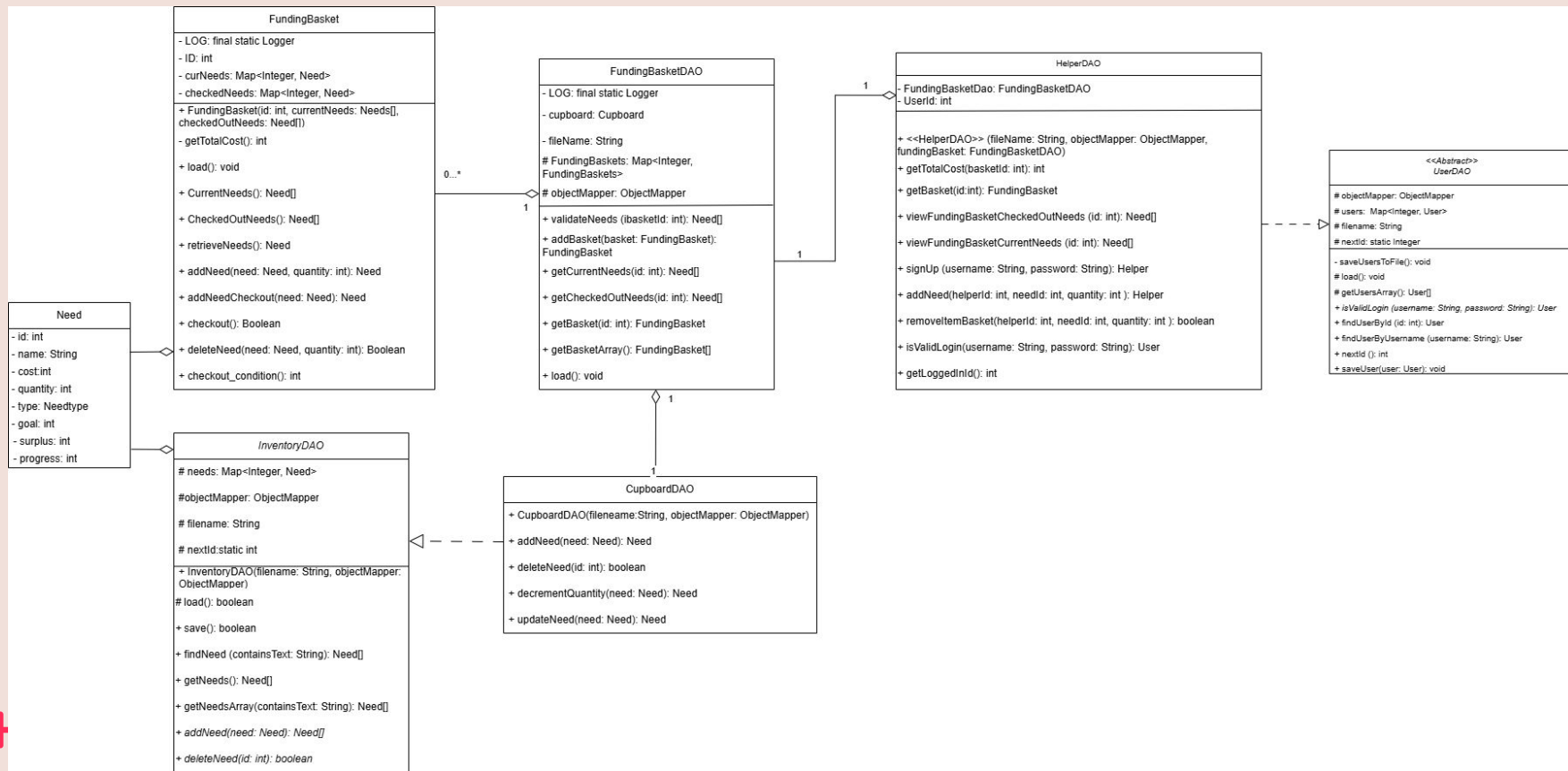
Initial design of the DAOs



Similar Design Structures



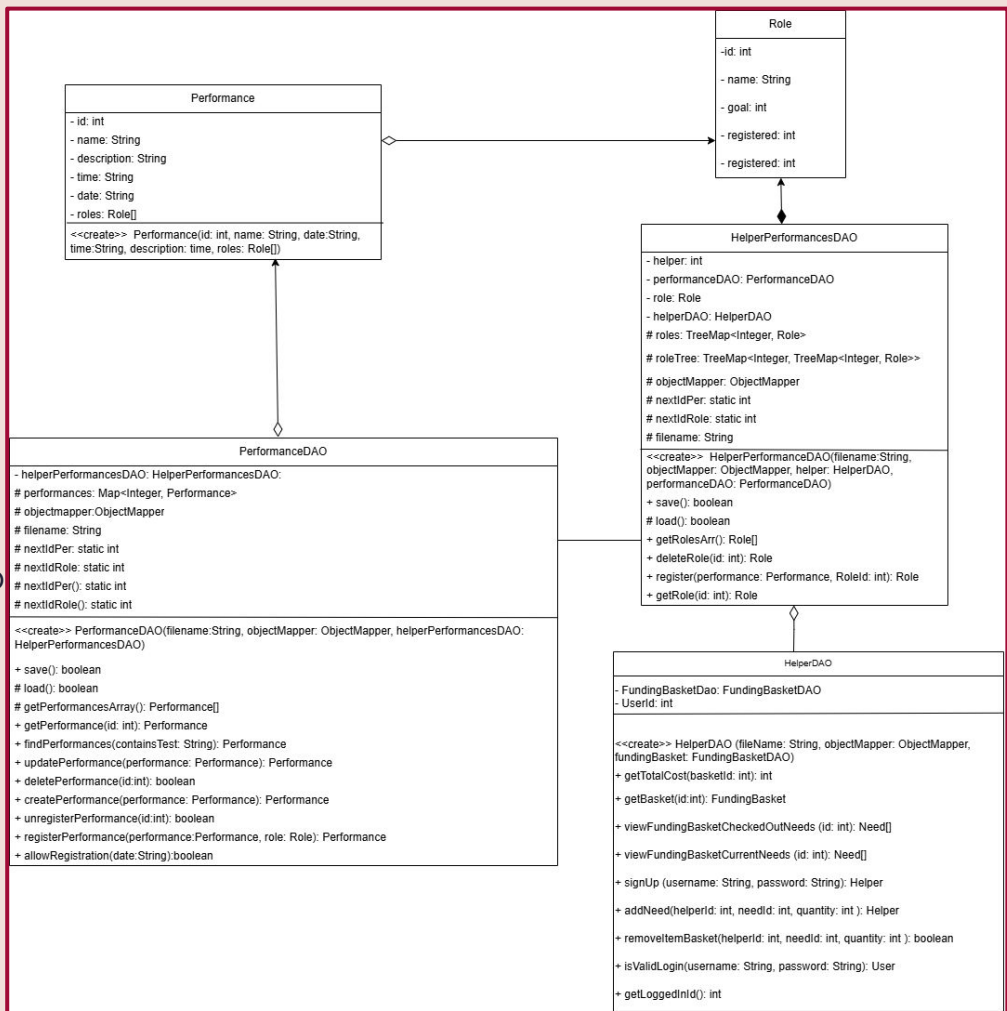
+ Helper Funding Basket UML





Domain Model Concert Enhancement

- Our Model Tier UML diagram of the performance enhancement we created
- Helper Performance DAO creates a new, “blank” Role object when first created (due to JSON object mapper)





Static Code Analysis highlights



☆ ufund-api / main ?

Overview Issues **Security Hotspots** Measures Code Activity

Project Settings ▾ Project Information

2 Security Hotspots

Review priority: ▾ Low

▾ Insecure Configuration 2 ▾

Make sure this debug feature is deactivated before delivering the code in production.

Make sure this debug feature is deactivated before delivering the code in production.

2 of 2 shown

↑

123 *
124 * @param need The {@link Need} object to be added to the basket.
125 * @param addBy quantity to addBy
126 * @return The added {@link Need} object if successfully added, or null if there was an error.
127 */
128 public Need addNeed(int helperId, Need need, int quantity) throws IOException {
129 try {
130 return fundingBasketDAO.addNeed(helperId, need, quantity);
131 } catch (IOException e) {
132
133 e.printStackTrace();
134
135 return null;
136 }
137 /**
138 * Remove an item from the funding basket.
139 *
140 * @param needId The ID of the need to be removed.
141 * @return true if the item was successfully removed, false otherwise.

Make sure this debug feature is deactivated before delivering the code in production.

For debugging purposes, we had been printing our stack trace in HelperDAO. However, in the future this could pose a security threat to our organization and website, giving malicious users access to sensitive vulnerability information, so this function will be deleted as part of routine maintenance.





Static Code Analysis highlights



Passed

New Code

Overall Code

UFund-API

Security

0 Open Issues

A

Reliability

15 Open Issues

C

Maintainability

637 Open Issues

A

Accepted issues

0

Valid issues that were not fixed

B

Coverage

74.1%

On 1.2k lines to cover.

C

Duplications

2.0%

On 4.7k lines.

D

Security Hotspots

2

E

SonarQube alerted us to some reliability issues within our code. Namely, that a few of our Controllers' variables do not have exactly-matching path names or do not bind properly with the URI links (GET, POST, PUT, DELETE, etc). This could cause serious issues down the line, with our product potentially failing to run because it cannot bind to these variables.



Many of our SonarQube flagged issues were in relation to obsolete and commented-out snippets of code that are just cluttering up our program.

SonarQube also told us that our code would be easier to maintain and safer from accidental breakage if we made the URLs within our services final variables as opposed to changeable global variables.



Quality Gate ⓘ

Passed

Last analysis 13 minutes ago

New Code

Overall Code

Heartstrings-app

Security

0 Open Issues

A

Reliability

4 Open Issues

E

Maintainability

237 Open Issues

A

Accepted issues

0

Valid issues that were not fixed

B

Coverage

0.0%

On 981 lines to cover.

C

Duplications

7.5%

On 6.7k lines.

D



Usability Improvements: Button Sizes and Format



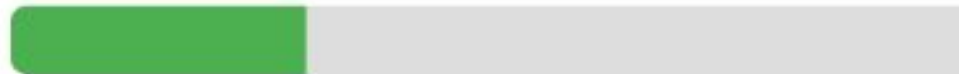
Name: Alto Saxophone

Cost: \$10.00

Quantity Still Needed: 88

Surplus: 0

Type: PHYSICAL



Add to funding basket:

go back

Need Name:

Cost

Need Goal:

Need Type: PHYSICAL

Go Back Save

Your total: \$20.00

Total Volunteer Hours: 0

Checkout

Go Back





Usability Improvements: Button Sizes and Format



Name: Alto Saxophone

Cost: \$10.00

Quantity Still Needed: 88

Surplus: 0

Type: PHYSICAL



Add to funding basket:

go back

Need Name:

Cost

Need Goal:

Need Type: PHYSICAL

Go Back Save

Your total: \$20.00

Total Volunteer Hours: 0

Checkout Go Back





User Interface Improvement: Manager Dashboard Router Links



DASHBOARD MANAGER

Logout

Performances

Home

DASHBOARD MANAGER

Logout

Performances

Home





Usability Improvement: Helper Dashboard - Funding Basket Accessibility



HELPER DASHBOARD

Logout

User Profile

Performances

Home

HELPER DASHBOARD

Logout

User Profile

Performances

Funding Basket

Home



+ UI Improvements: Searching for Needs



Search Needs

- [Alto Saxophone](#)

Search Needs

- **Alto Saxophone**

Search Needs

- **Alto Saxophone**



What went well?

- The cupboard system
- Maintaining our UI architecture and design system
- Achieving MVP+
- Performances enhancement looks and work great!

What didn't?



- Timing + underestimating how long we spent working
- Maybe more improvements on accessibility ie. dark vs light mode



+

THANK YOU!

+

