Donation Web Application

Analysis and Design Document

Roman Bogdan-Ovidiu

**30431**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

# Project Specification

DWA is a Web Application which will be used to help people to donate what they want easier and faster. There are three types of users, guest, who can see the items and write a request for them, logged in users who can post items they want to donate, and admins who can edit /remove etc. items ,members .

A member post an item for donation, and others can just make a request with their story(why they need the item for) after the donor see the requests , he chooses to who should his item go to.

So the user can perform: - log in : If logged in : -add an item , wait for requests, give the item!

-same as not logged in user ->

* Not logged in: find an item you would need, write to the user a story why would you need that item and wait until he decides if you deserve it.
* Admin: check the web for spam, for stupid items etc. and remove them .
* Number of viewers dynamically on the screen etc.

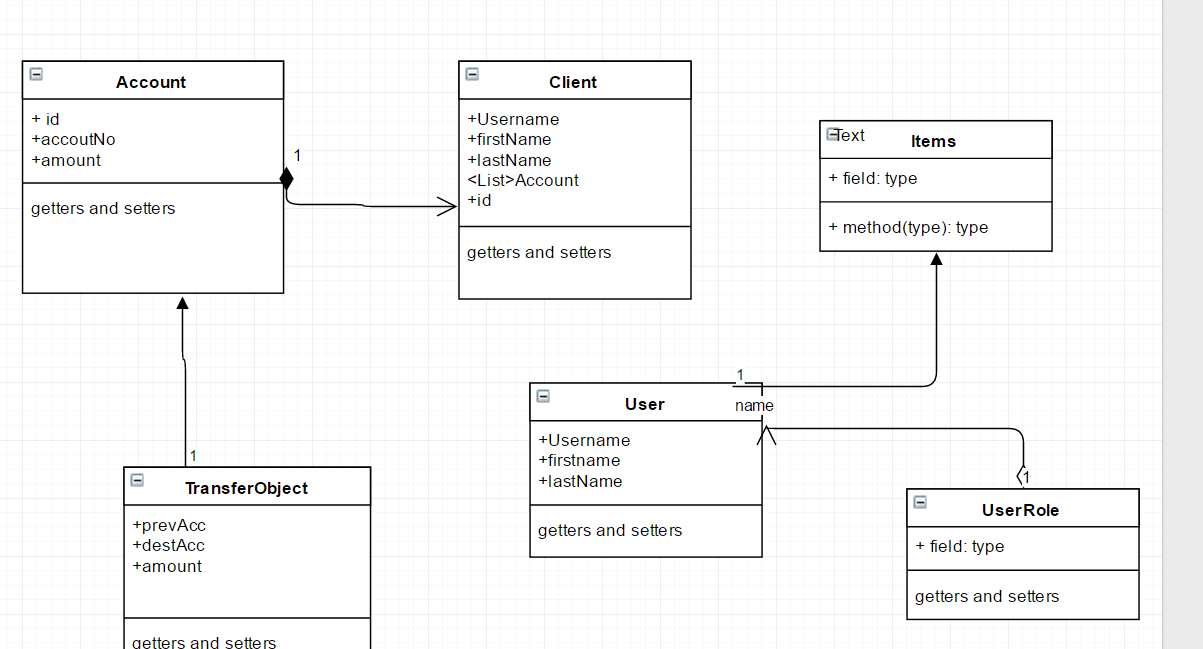
Technologies: Spring,MVC,Hibernate,Maven,Mysql,Java.

User Interface : will be created in JSP , main page having 2 buttons , create new account, log in if you already have and account or go to products which will let you search through products but you won’t be able to apply for them only if you are logged in. Each button will redirect to the desired page.

# Elaboration – Iteration 1.1

# Domain Model

The model will have the following entities: User- Admin and Clients, Account for transfers, Items which will be donated.



# Architectural Design

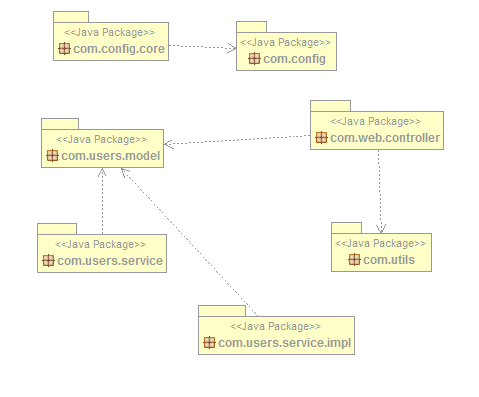
## Conceptual Architecture

I will use Layers Architecture and MVC Architecture pattern. Common principles for designs that use the layered architectural style include:

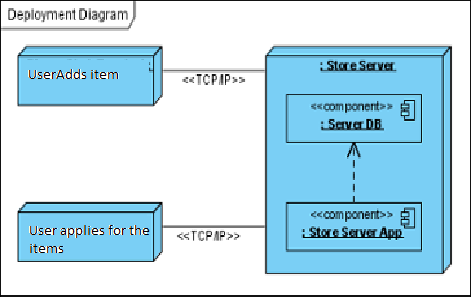
* **Abstraction**. Layered architecture abstracts the view of the system as whole while providing enough detail to understand the roles and responsibilities of individual layers and the relationship between them.
* **Encapsulation**. No assumptions need to be made about data types, methods and properties, or implementation during design, as these features are not exposed at layer boundaries.
* **Clearly defined functional layers**. The separation between functionality in each layer is clear. Upper layers such as the presentation layer send commands to lower layers, such as the business and data layers, and may react to events in these layers, allowing data to flow both up and down between the layers.
* **High cohesion**. Well-defined responsibility boundaries for each layer, and ensuring that each layer contains functionality directly related to the tasks of that layer, will help to maximize cohesion within the layer.
* **Reusable**. Lower layers have no dependencies on higher layers, potentially allowing them to be reusable in other scenarios.
* **Loose coupling**. Communication between layers is based on abstraction and events to provide loose coupling between layers.

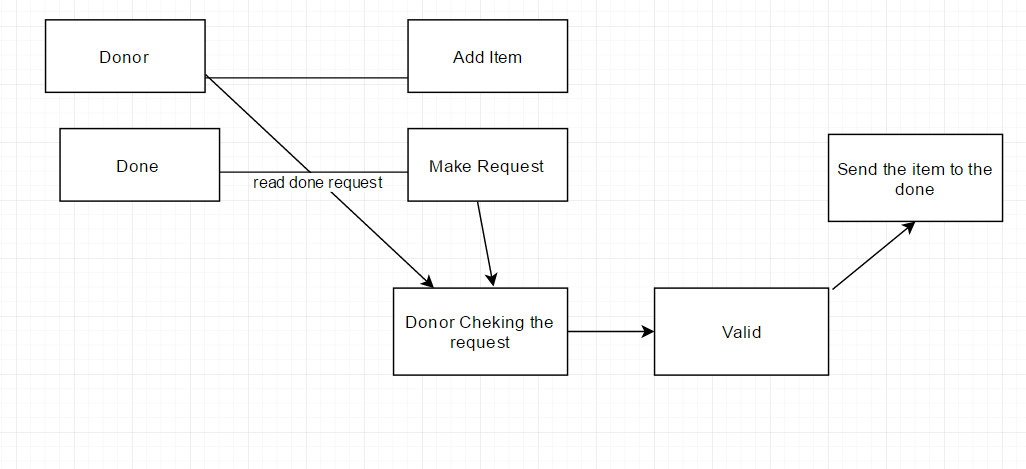
## Package Design

There will be the following packages: com.model with the entities/models, com.repository with the repositories, com.controller with the GUI linking and controlling the functionality of the Jsps , com.service with the interfaces of the service, and com.serviceImpl with the implementation of the service interfaces.



## Component and Deployment Diagrams

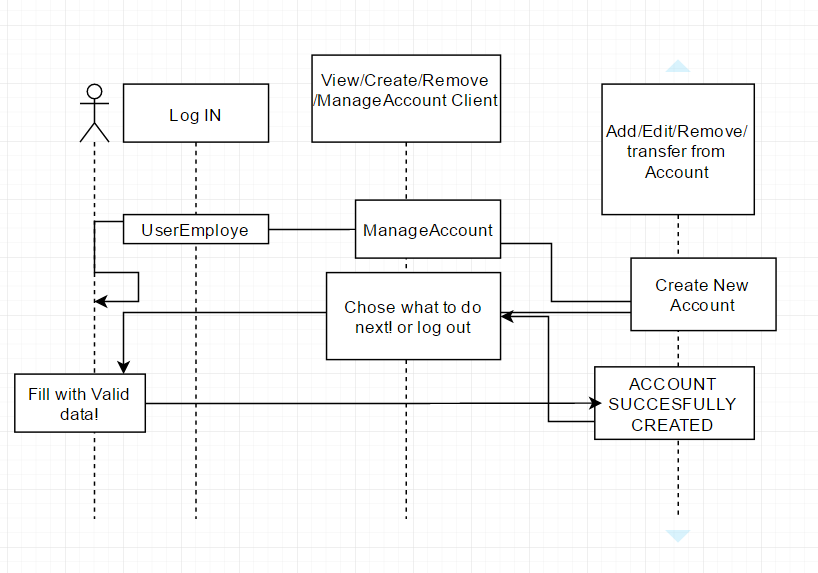




# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior



## Class Design

# 

# Data Model

# Unit Testing

The project was tested manually.Only some functions where tested using JUnit .

# Elaboration – Iteration 2

# Architectural Design Refinement

### MVC design pattern

This design pattern helps us develop loosely coupled application by segregating various concerns into different layers. MVC design pattern enforces the application to be divided into three layers, Model, View and Controller.

**Model:** This represents the application data.

**View:** This represents the application’s user interface. View takes model as the input and renders it appropriately to the end user.

**Controller:** The controller is responsible for handling the request and generating the model and selecting the appropriate view for the request.

# Construction and Transition

# System Testing

The system was tested manually , every function and every button.

# Future improvements

As future improvements I propose to:

-display image from database, not from path

-make email sender work(its implemented but it has a bug)

# Bibliography

<http://www.codejava.net/frameworks/spring/understanding-spring-mvc?showall=&start=1#mvc>

<http://www.tutorialspoint.com/spring/constructor_based_dependency_injection.htm>

http://docs.spring.io/spring/docs/current/spring-framework-reference/html/mail.html