Mercado Gravy
Software Architecture Document

Version <1.0>

Revision History

Date	Version	Description	Author
04/10/2018	1.0	SAD first version	Eva van Esseveldt, Jimena Lozano, Maite Herran, Pedro Pingarilho, Fermin Gomez, Emilio Basualdo

Table of Contents

Software Architecture Document

1. Introduction

1.1 Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

1.2 Scope

The following document exposes the main decisions taken to define the principal structures used. A description of each structure together with the essential specifications is also provided.

1.3 Definitions, Acronyms, and Abbreviations

See glossary set out in the project's principal folder.

1.4 References

This Software Architecture Document references solely to the project's repository:

- https://github.com/maiteherran/ingenieria-de-software

1.5 Overview

The rest of the Software Architecture Document exhibits the objectives and restrictions of the architecture, following the main features that identifies it together with system expectations such as its quality.

2. Architectural Representation

The views necessary for the current system are:

- The home view
- The register view
- The login view
- The profile view
- The create item view
- The cart view
- The item view

3. Architectural Goals and Constraints

The architecture has been designed with the following objectives:

- To provide a secure place where people can trade their goods.
- To make the platform be as simple as possible for users so that they can have a good experience.
- To ensure privacy for users so that they can trust their payment information to the system in order to make transactions.
- Regarding the code, an object oriented one.

The major constraint is how each of the members of the team skills affect in the viability of the project and the time it takes for it to be fulfilled.

4. Use-Case View

Due to the fact that the project is aimed for people, almost all of the use cases display an interaction between the user and the platform with the exception of one which shows the interaction between the platform and the database. Search for Item, See Cart and List of Products use cases may show data to the user or, if there's no data to display or an error occurs, let the user know through a message.
In Sign Up, Log In, Edit, Delete Account, Add to Cart, Edit Cart, Checkout, Verify Payment, Place Item and Edit List use cases, the system may inform if errors were made or if the information provided is valid.

5. Logical View

The main actors of the model are the users, the user administrator and the system where information is manipulated and shown to the other actors. Users include sellers who can list items with their pertinent information and buyers who are able to buy items. Both buyers and sellers can modify their own information as users. The user administrator interacts with the system as a moderator to manipulate the users and what they can post or not.
A detailed class diagram is shown in the folder i*ngenieria-de-software/Documentación Técnica/Diagrama de clases*

5.1 Overview

Not applicable.

5.2 Architecturally Significant Design Packages

Not applicable.

5.3 Use-Case Realizations

Fourteen different use cases are represented with their corresponding activity diagrams inside the folder *ingenieria-de-software/Documentación Técnica/Casos de Uso.*

6. Process View

Multi threads will be implemented in order to allow concurrent execution of two or more parts of the program. Processes will communicate via exceptions and object messages.

7. Deployment View

If the project were to be implemented on a web basis, the nodes involved would be those of the the administrator that manages the users, the sellers and buyers (users),and that node where the site is hosted. However, for the project's scope, we are simply looking for a single computer to be able to accurately represent the platform.

8. Implementation View

The present product's software will be implemented using Java as the programming language and PostreSQL for the database .

8.1 Overview

Not applicable.

8.2 Layers

Not applicable

9. Data View (optional)

User data will be saved (this includes their user information and the items they post) in an architecture which assures a fast access to it. Space complexity will not be prioritized as it isn't highly relevant in the present project.

10. Size and Performance

The architecture will be one that will allow simple and fast software for the user.

11. Quality

Reliability is the main capability of the system that the software architecture focuses on due to the fact that users will be introducing personal important information. Also, an efficient object oriented design will guarantee that new features could be easily added in the future.