Rasti & Co.

Kaufen
Software Architecture Document

Version <1.0>

Revision History

Date	Version	Description	Author
03/05/2018	1.0	First version	Agustina Osimani Micaela Banfi Clara Guzzetti

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 Software Architecture Document outlines the decisions that lead to the principal structures used. Furthermore, it gives a description of each structure, including the fundamental specifications that allow the comprehension of the system in its environment.

1.3 Definitions, Acronyms, and Abbreviations

N/A

1.4 References

Kaufen project: https://github.com/nacho9900/IngeSoft

1.5 Overview

The structure of the Software Architecture Document consists in three main parts. First, it shows the architectural goals and constraints. Following that, the main aspects of the architecture are listed and developed. This aspects include the Use-Case, Logical, Process, Deployment, Implementation and Data.

Finally, it details the expectations of Quality, Size and Performance of the system.

2. Architectural Representation

The views that are necessary to describe the software architecture and its representation are:

- Home Screen
- Search page
- Cart page
- Purchase page
- User profile

The underlying software used for modeling the views above is the free platform draw.io.

The models of each view can be found in the git of the project, in IngeSoft / Casos de Uso / Maquetas.

3. Architectural Goals and Constraints

Maximum security is required in all areas concerning monetary transactions, including but not limited to

credit cards and bank transactions. The privacy of users should be guaranteed as they trust their credit information to the system in order to carry out the transactions.

The usability of the system is sought to provide the best user experience.

As for the code, it is expected portability. Due to that, it should be object oriented, ensuring reusability.

4. Use-Case View

Given the nature of the project, most use-cases show the interaction between the user and the platform. In case user information is required, the system informs whether it is valid or if errors were made. In the other hand, when information is displayed to the user, the system shows the information in case it exists, or inform an error occurred. This one may be caused because the user does not have items for sell to be shown.

Central functionality of the final system can be seen in the following use-cases:

- Use_Case_01: User registration
- Use_Case_04: Product searching
- Use_Case_06: Adding product to cart
- Use_Case_10: Product ABM

5. Logical View

The class diagram is shown in the git of the project, in IngeSoft / Documentacion Tecnica / Diagrama de Clases.pdf.

Within the model, there are two main actors, the User, and the AdminUser. Users can provide products to sell, uploading pictures and information of each one. Moreover, they can purchase products from other users and they can modify their personal information. The AdminUser interacts with the system to manipulate existing Users.

The model has fundamental classes regarding user experience, such as Cart, Product and Review.

There system will also have a view that contain all the visual elements that are displayed in the page.

Finally, there is the controller, which will handle all the interactions between the view and the model. It will also be in charge of maintaining the database updated when modifications are made to the model.

5.1 Overview

N/A

5.2 Architecturally Significant Design Packages

The system will be developed utilizing a MVC oriented development strategy, thus Model, View and Controller will be separated from each other within their own packages.

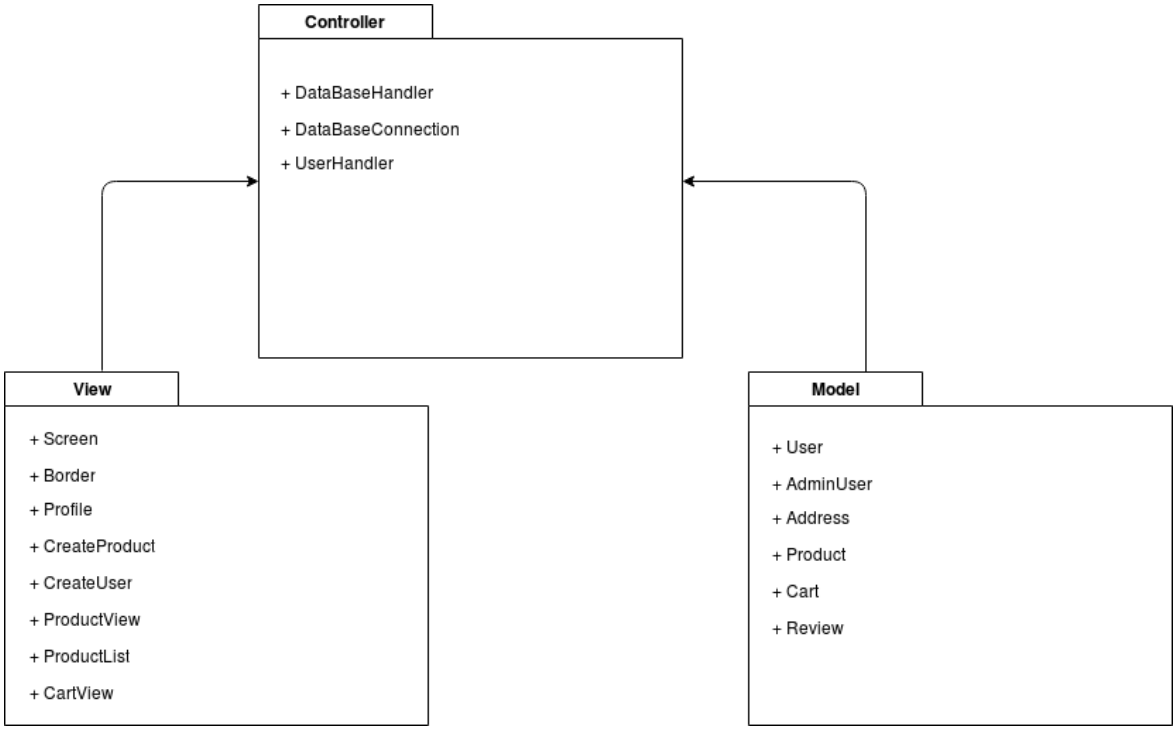

Figure 1: Packages Diagram

5.3 Use-Case Realizations

Twelve use-cases were taken into account in the realization of the software. They are specified, including a diagram for each one, in the git of the project, in IngeSoft / Casos de Uso / Uses Cases - Diagrama de Casos de Uso y Fichas.pdf.

6. Process View

Object Messages and exception, via the Java Virtual Machine, will derived the main flow of the application.It will maintain a multi-thread architecture.

7. Deployment View

The software will be able to run through a 32-bit or 64-bit installation of the Java Virtual Machine interconnected to a SQL database through a remote database protocol.

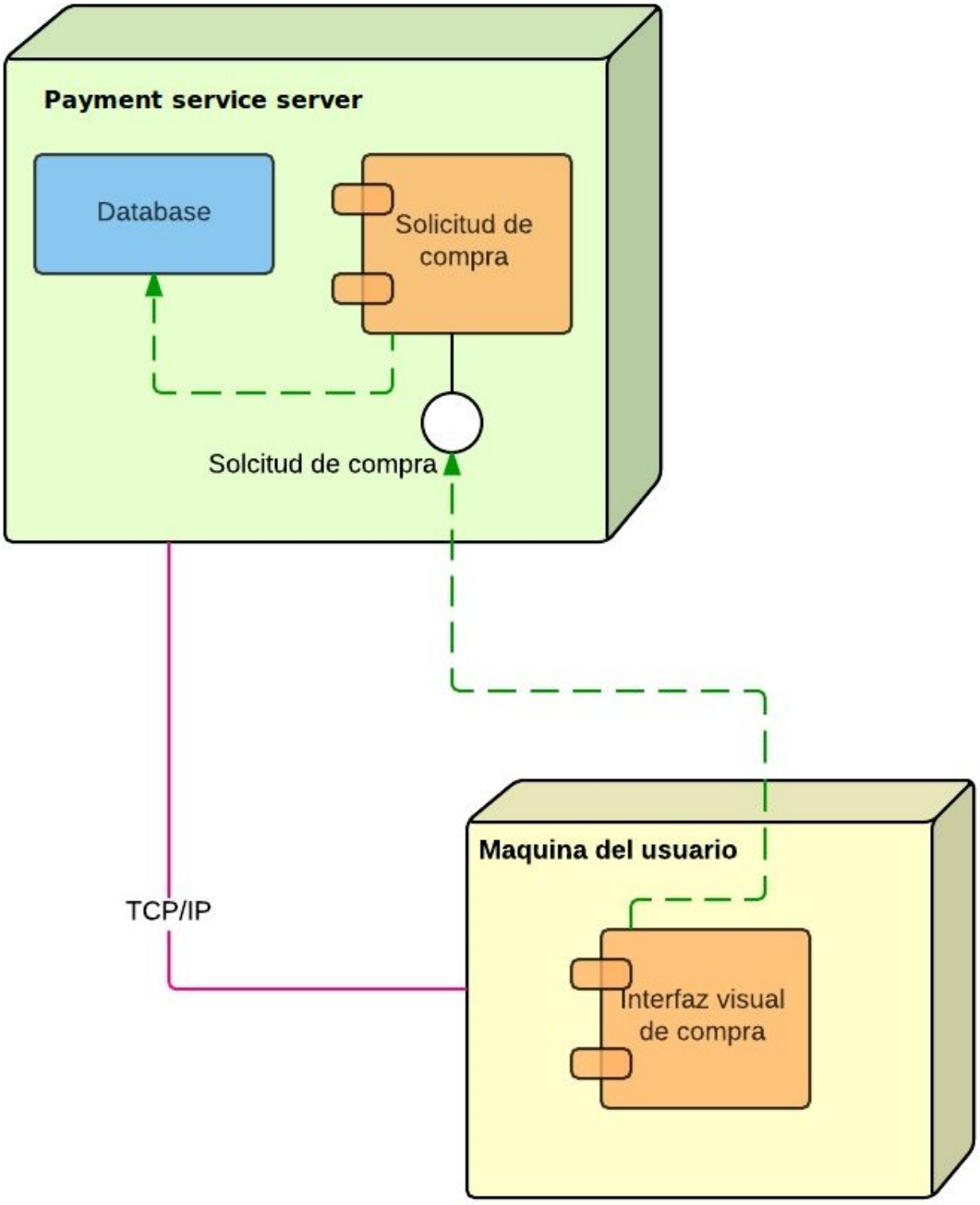

Figure 2: Components and Deployment

8. Implementation View

The software will be written in the Java Programming Language, Swing FX for the GUI and Postgresql for

the database.

8.1 Overview

N/A

8.2 Layers

N/A

9. Data View (optional)

User and items data will be stored. As the spatial complexity would not be prioritized, this data storage is not seen as an important matter. The items will be stored in the most convenient data structure available.

10. Size and Performance

Kaufen App will have a fast and efficient software so that both, buyer and seller, can upload products or buy them as satisfying and instantly as possible.

11. Quality

The primary concern in the architecture of Kaufen app will be reliability since our customers will interact with personal information such as credit card or bank account numbers. Also, in order to increase usability and satisfy the users demands, a special focus in the user design and portability will be made. Further on, a good object oriented design, will guarantee the addition of new features easily.