Supermarket Manager

Analysis and Design Document

Student: Marin Andreea

**Group: 30431**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 10/04/23 | 1.0 | First version | Marin Andreea |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

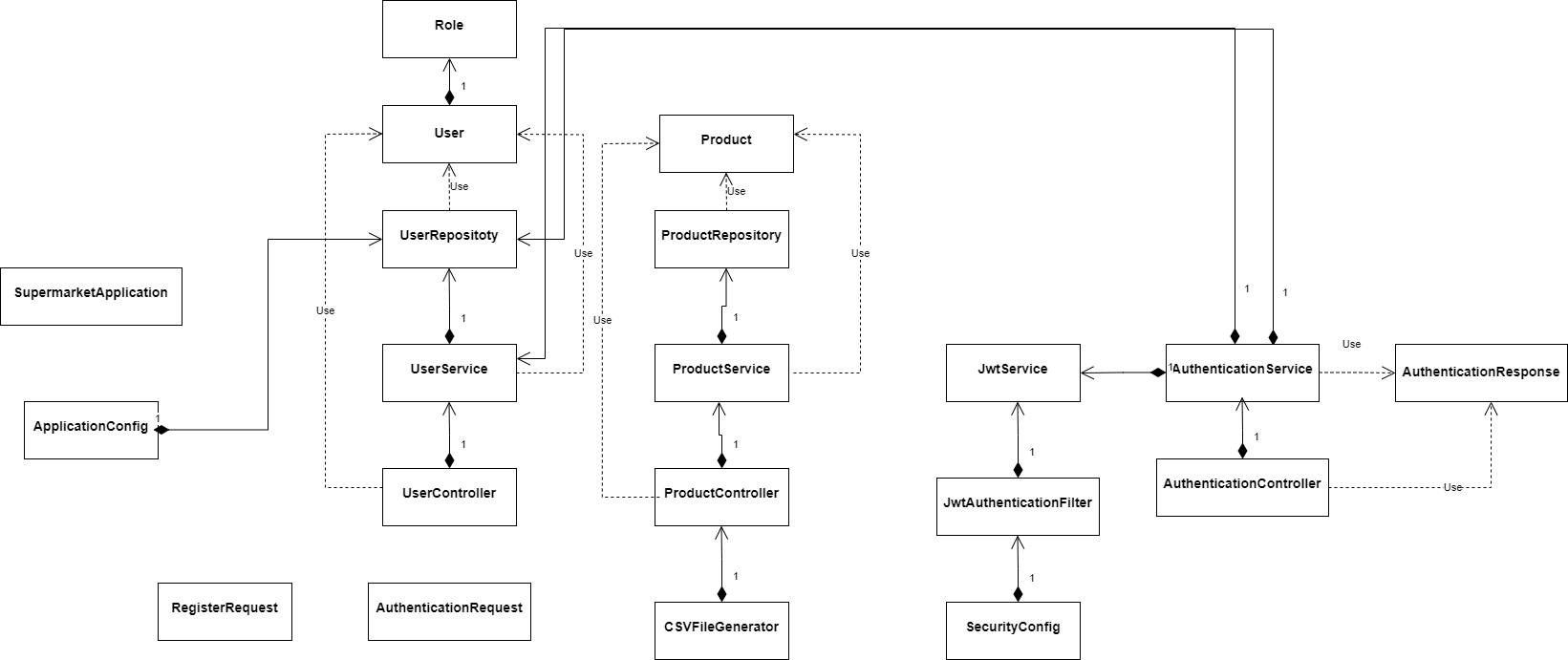
VI. Bibliography 5

# Project Specification

The goal of this project is to design and implement a client-server application for the employees of a supermarket. The application will have two types of users (a regular user represented by the cashier, and an administrator user). Both of them have to provide a username and a password in order to use the application.

# Elaboration – Iteration 1.1

# Domain Model



The conceptual class diagram of the project presents the classes contained by my project and illustrates the relationships between them. “Role” is an enumeration class that contains two strings: ADMIN and CASHIER and it is used to make the difference between the users of the application. UserRepository and ProductRepository are interfaces that extend **CrudRepository** and are used for implementing CRUD operations.

The authentication is done using jwt tokens. When the user logs in the application a token is generated and then is used for authorization.

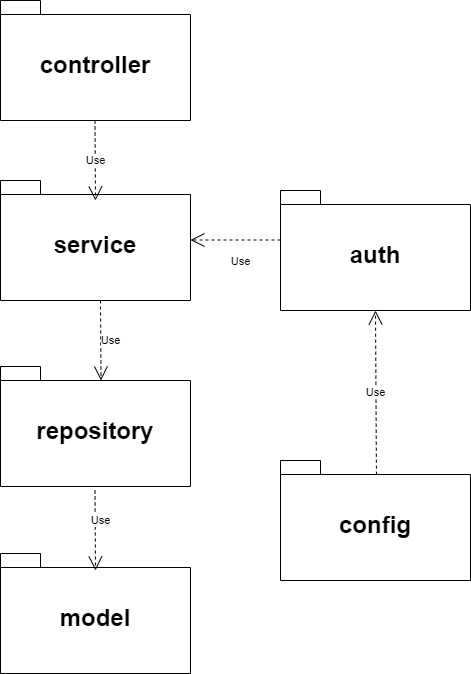
# Architectural Design

## Conceptual Architecture

The architecture used for this project is a layered one, as it can be deduced from the class diagram. Layered architecture is said to be the most common and widely used architectural framework in software development. It is also known as an n-tier architecture and describes an architectural pattern composed of several separate horizontal layers (in my case model, controller, repository, service) that function together as a single unit of software. I chose this architectural style because layers are only connected to the layers directly below them which means that layers can be modified and the change won’t affect other layers.

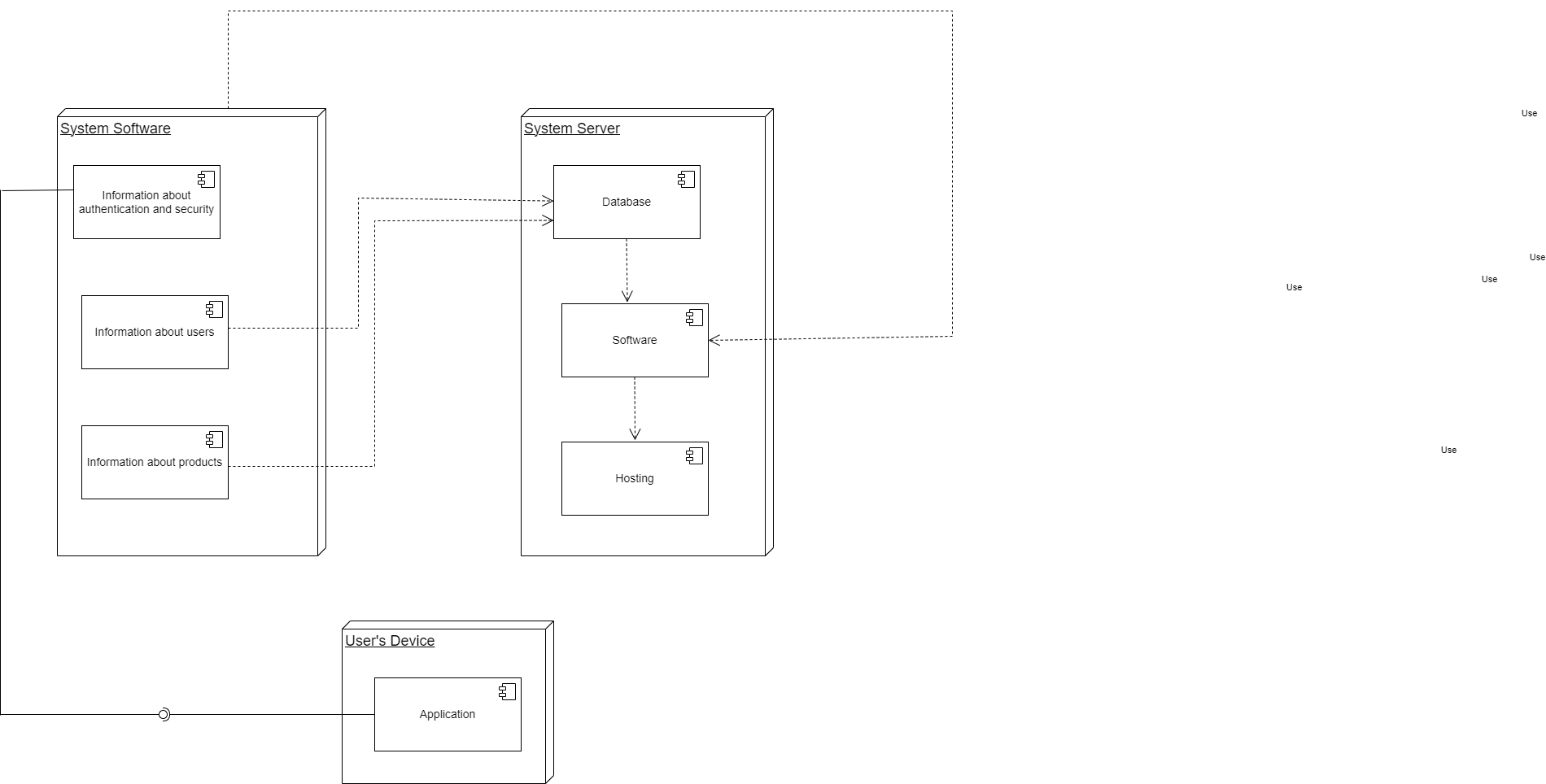
## Package Design

The package diagram also highlights the use of a layered architecture.

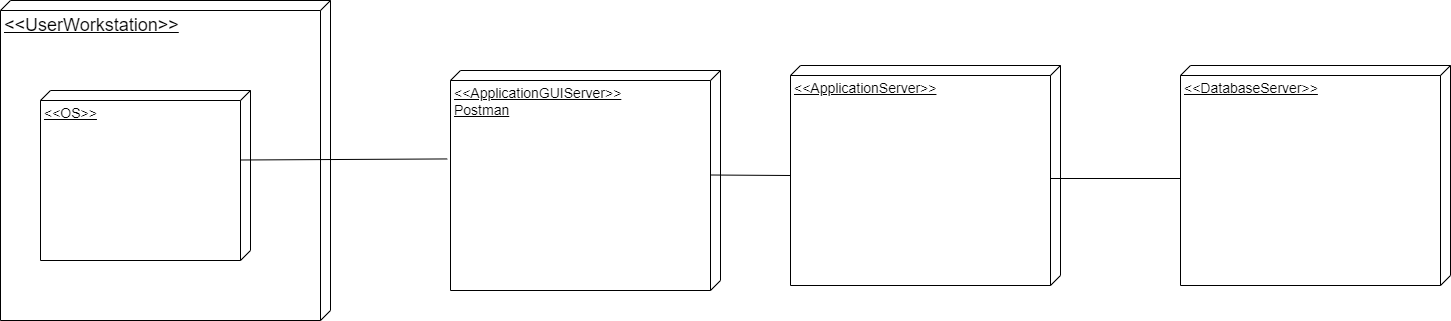


## Component and Deployment Diagrams

Component Diagram



Deployment Diagram



# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

*[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]*

## Class Design

*[Create the UML class diagram; apply GoF patterns and motivate your choice]*

# Data Model

*[Create the data model for the system.]*

# Unit Testing

*[Present the used testing methods and the associated test case scenarios.]*

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

*[Describe how you applied integration testing and present the associated test case scenarios.]*

# Future improvements

*[Present future improvements for the system]*

# Bibliography