<Online Grocery Shopping >

Analysis and Design Document

Student: Antonescu Maria-Cristina

**Group: 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

# *The scope of the project is to design and develop a web application that will enable customers to purchase groceries online and have them delivered to their doorstep. The admin will have access to the backend of the application, where they can perform CRUD (Create, Read, Update, Delete) operations on the products and manage orders.*

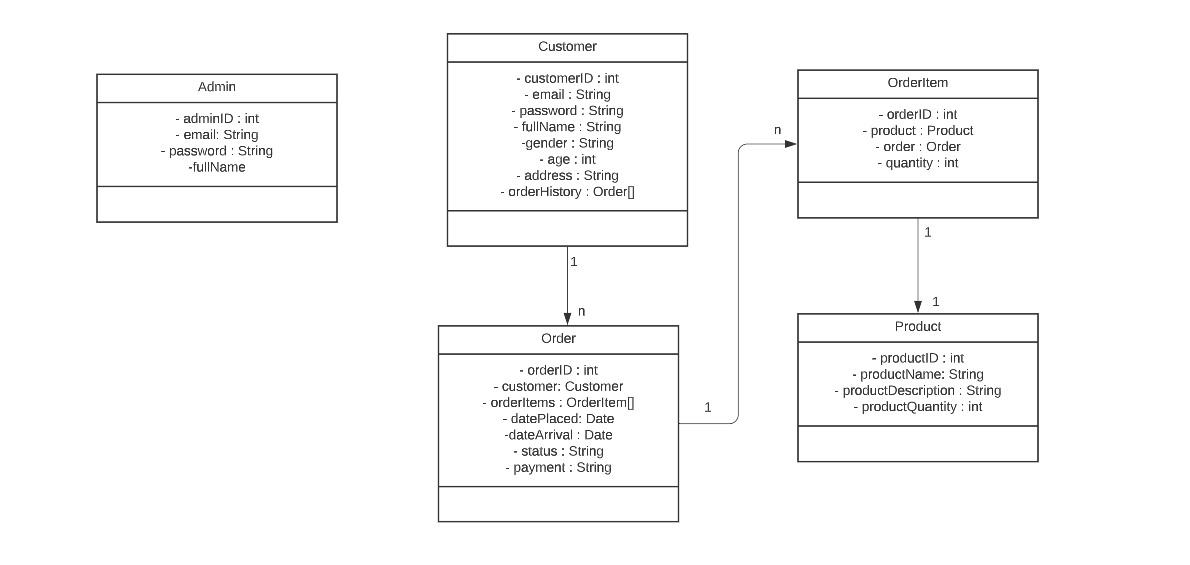
# Elaboration – Iteration 1.1

# Domain Model

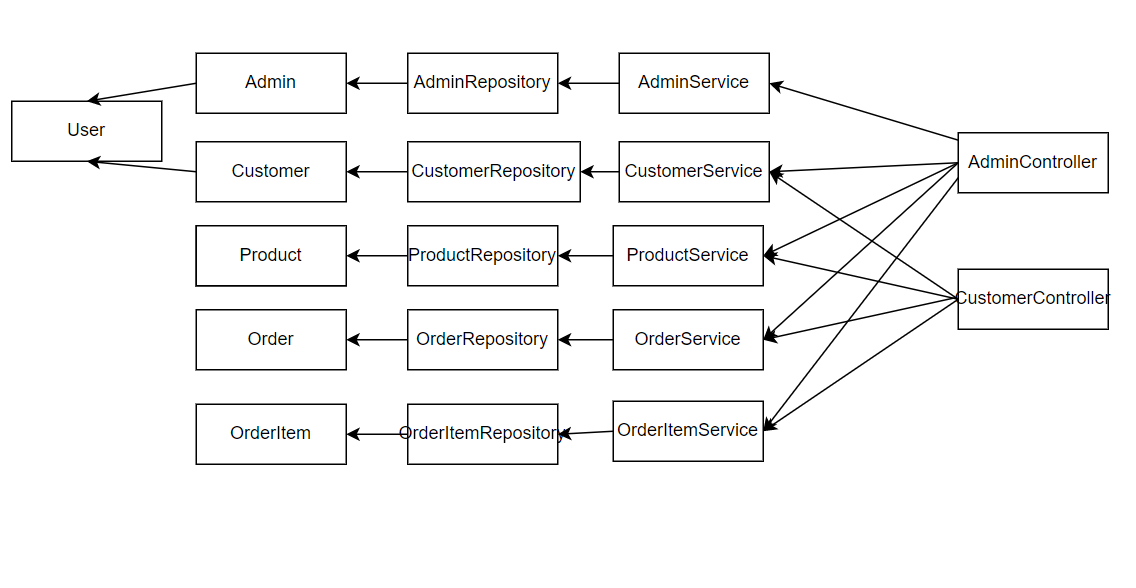
*The main entities in the system will be the following:*

* *Admin – a type of user, which can log-in the app, perform CRUD operations Customers, Products and Order;*
* *Customer – can register, log-in the app, can place an Order, view products;*
* *Product – described by the unique code, name, description and current quantity;*
* *Order – described by customerID, other details such as the date it has been placed, the date it should be delivered, address, the order items and the payment details*
* *OrderItems – entity which manages the relationship between Product and Order, it has productid, quantity, orderID.*

*The domain model:*

**

*The conceptual class diagram:*

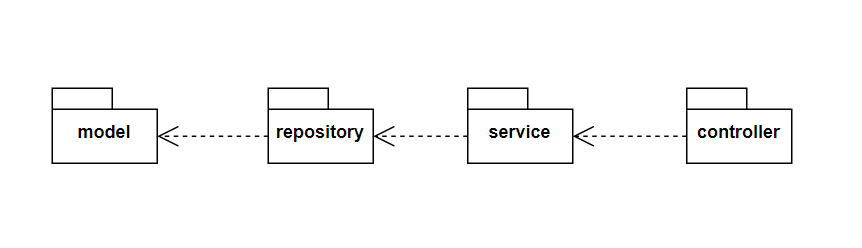
**

# Architectural Design

## Conceptual Architecture

1. *Presentation layer: This layer is responsible for handling user requests and responses. It consists ina controller that processes user input and returns the appropriate response.*
2. *Service layer: This layer is responsible for handling business logic and performing data manipulation. It acts as a bridge between the presentation layer and the data access layer.*
3. *Data access layer: This layer is responsible for accessing data from a database or other data storage system. It consists of data access objects (DAOs) that perform CRUD (Create, Read, Update, Delete) operations on the data.*

## Package Design

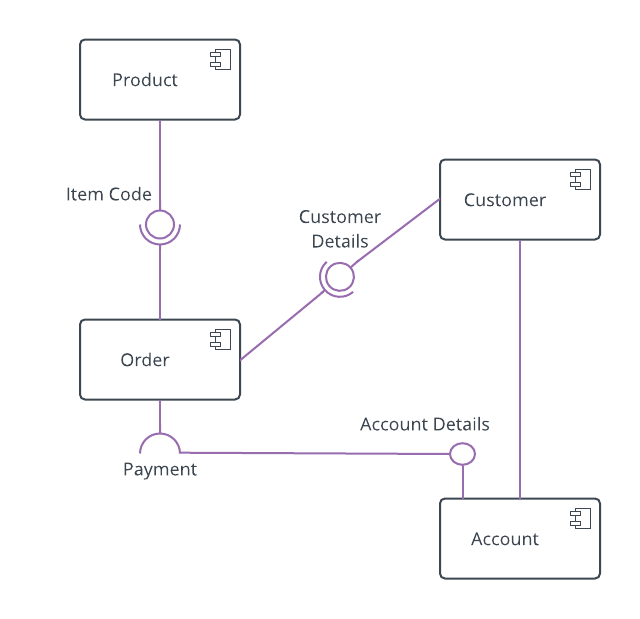
**

*Package description:*

* *Model – the package which contains the entity definitions*
* *Repository – the package which contains the JpaRepositories*
* *Service – the package which contains the Service classes*
* *Controller – the controller for the 2 types of users: Admin and Customer*

## 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