Grocery Shopping Android Application

Analysis and Design Document

Cordea Corina

**Group 30432**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 04/04/2018 | 1.0 | Domain Model and Architectural Design | Cordea Corina |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 5

2.1 Conceptual Architecture 5

2.2 Package Design 6

2.3 Component and Deployment Diagrams 6

III. Elaboration – Iteration 1.2 7

1. Design Model 7

1.1 Dynamic Behavior 7

1.2 Class Design 7

2. Data Model 7

3. Unit Testing 7

IV. Elaboration – Iteration 2 7

1. Architectural Design Refinement 7

2. Design Model Refinement 7

V. Construction and Transition 7

1. System Testing 7

2. Future improvements 7

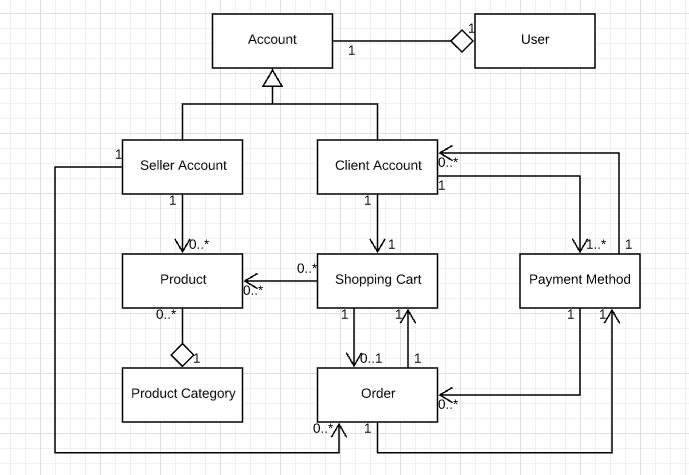
VI. Bibliography 7

# Project Specification

The Grocery Shopping is an android application where clients can purchase and order groceries online. The system is developed with a user-friendly and attractive GUI. Users have to first login into the system to view the groceries and add them into their cart. They can then order it by choosing a payment method. Also, the sellers can use the application to add their products and sell them online.

# Elaboration – Iteration 1.1

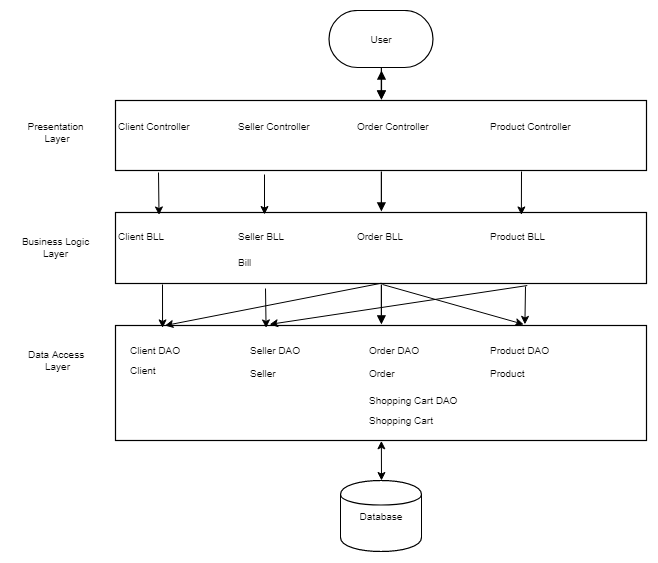
# Domain Model

**

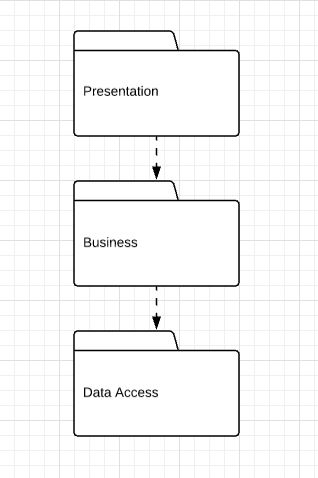
# Architectural Design

## Conceptual Architecture

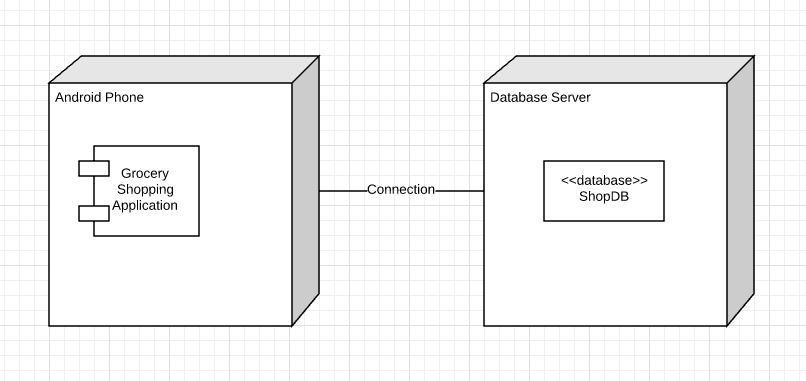
The Layers architectural pattern will be used to structure the application by dividing it into groups of subtasks based on their functional responsibility. More specifically, the application is divided in 3 layers (bottom to top): data layer which provides access to the data stored in a database, business logic layer which implements the main functionality of the system and presentation layer which contains the user oriented functionality and manages the interaction between the user and the system. Each layer can only access the one beneath it. By using this pattern, the maintainability of the application and the reusability of components are considerably increased.



## Package Design

**

## Component and Deployment Diagrams



# 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