Shopping Cart Service Design Document

BOOTCAMP JAVA 2016

2016

Table of Contents

[1 Introduction 2](#_Toc441423657)

[1.1 Purpose 2](#_Toc441423658)

[1.2 Scope 2](#_Toc441423659)

[1.3 Definitions, Acronyms and Abbreviations 2](#_Toc441423660)

[2 Design Overview 3](#_Toc441423661)

[2.1 Introduction 3](#_Toc441423662)

[2.2 Services Design 3](#_Toc441423663)

[2.3 Services Interfaces 3](#_Toc441423664)

[2.3.1 External User Interface Requirements 3](#_Toc441423665)

[3 Object Descriptions 4](#_Toc441423666)

[3.1 Objects 4](#_Toc441423667)

# Introduction

This document describes all the details necessary for the implementation of the specified services. The first sections portray the purpose and scope of the document. What is more, the process that services perform to satisfice the requirements.

## Purpose

The aim of this document is to specify, document and provide a detailed description of how services work and their interface.

## Scope

This document is for a base level system which will work as a proof of concept for the use of building a set of service. It will affect all the stages of the development process: analysis, implementation and testing.

## Definitions, Acronyms and Abbreviations

UML: Unified Modeling Language

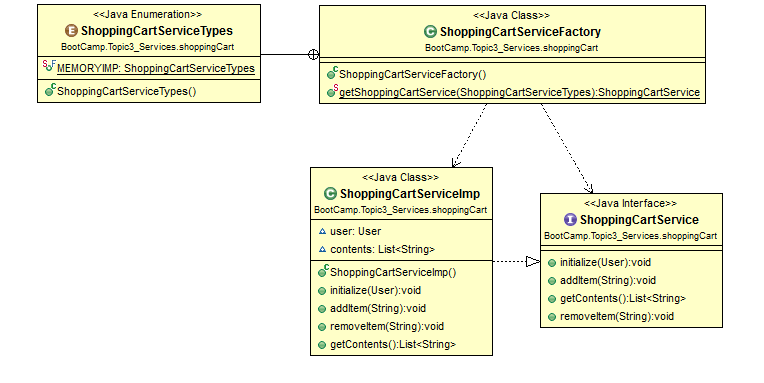
# Design Overview

## Introduction

The Design Overview is a section to introduce and give a brief overview of the design. The Services Design is a way to give the overall view of a system and to place it into context with external systems. This allows for the reader and user of the document to orient themselves to the design and see a summary before proceeding into the details of the design.

## Services Design

UML class diagram of the inherent classes in the services.



## Services Interfaces

### External User Interface Requirements

#### User Interfaces

The user interface for the system will allow the user to easily manage their shopping cart service by adding and removing items from it, as well as get the content of their current cart.

* addItem: Adds an item to the cart.
* removeItem: Removes an item from the cart.
* getContents: Obtains the contents of the current cart.

#### Software Interfaces

The software would need to interface with a persistent data management system to get data from it. The connection would be a standard database connection using JDBC or ODBC, although to simplify this example services will only manage data memory.

#### Communication Interfaces

The software would need to interface with a persistent data management system to get data from it. The connection would be a standard database connection using JDBC or ODBC, although to simplify this example services will only manage data memory.

# Object Descriptions

## Objects

|  |  |
| --- | --- |
| **Class name:** ShoppingCartServiceFactory | |
| **Brief description:** This class is responsible for creating the appropriate shopping service object that will be used. | |
| **Methods (operations)** | **Method Description** |
| getShoppingCartService  (ShoppingCartServiceType); | This method returns the right ShoppingCartService implementation based on the ShoppingCartServiceType that it gets by parameter. |

|  |  |
| --- | --- |
| **Interface name:** ShoppingCartService | |
| **Brief description:** This interface defines the operations that every ShoppingCartService should implement. | |
| **Methods (operations)** | **Method Description** |
| initialize(User) | Initializes the service and defines the user owner of the cart. |
| addItem(String) | Adds an item to the Cart. |
| removeItem(String) | Removes a specified item from the Cart. |
| getContents() | Returns the current content from the Cart. |

|  |  |
| --- | --- |
| **Class name:** ShoppingCartServiceImp | |
| **Brief description:** This class is responsible for implementing the services provided for the ShoppingCartService interface. This class uses a list to store the items added to the cart. | |
| **Attributes (fields)** | **Attribute Description** |
| user:User | User who is using the service and owner of the cart. |
| contents:List<String> | List which represents the items added to the cart. |
| **Methods (operations)** | **Method Description** |
| initialize(User) | Initializes the service and defines the user owner of the cart. |
| addItem(String) | Adds an item to the list. |
| removeItem(String) | Removes a specified item from the list. |
| getContents() | Returns the current content from the list. |

|  |
| --- |
| **Enumerator name:** ShoppingCartServiceTypes |
| **Brief description:** This enumerator enumerates the different implementations of the ShoppingCartService interface in order to be created by the factory |