System Design Document for Inventory System Development

Shu Ou -- UCSD Extension Java III

System Elements Overview

The Inventory System consists of a total of one interface, eight classes and one enum as listed Below:

- ISMain
- InventorySystem
- InventoryMaintenance
    - Book
    - Cd
    - Dvd
- ISCat - enum
- IS Model
- ISView
- ISController

Detailed Descriptions of System Elements

ISMain

To simply put it, the ISMain initialize the inventory system by making used of the itemselection() method, is the connector of the ISModel Model Component), ISView (View Component), and ISController (Controller component).

InventorySystem

The InventorySystem interface contains general methods for actions to get property (getProps()), add property (addProps()), search property (searchProps()), delete property (deleteProps()), and edit property (editProps()). Such methods are overridden by the InventoryMaintenance class for more defined usage.

InventoryMaintenance

The InventoryMaintenace class implements the InventorySystem interface and overrides all InventorySystem methods as stated above except for the searchProps() method. In addition, such classes also contain five methods to breakout information for different fields of an item (Table 1). For example, the getFieldOne method will extract information for what info is currently in field one of an item.

- **Get Properties**: load current inventory, display inventory
- **Add Properties:** search the item in which the user wants to add, inform user if item is already in inventory; if not, add item to inventory
- **Delete Properties:** displace the current inventory to user and let user decide which item s/he would like to delete
- **Edit Properties:** Replace old information with new information for an item based on the item title and field number provided by the user
- **Get Fields (getFieldOne, getFieldTwo…):** Fields in this inventory system is separated by a forward slash "/". These methods will be able to extract individual field info based on the "/" number.
- **Table 1**

Category	Field 1	Field 2	Field 3	Field 4	Field 5
Book	Title	Author	Price	Published Year	Rating
CD	Title	Artist	Price	Released Year	Rating
DVD	Title	Director	Price	Released Year	Rating

Book, Cd, Dvd

These three classes are subclasses for the InventoryMaintenance class. Each one of them implements a slightly different search Properties method based on the its own fields.

- Search Properties: User provides item title for which s/he wants to search. Method search the title within inventory, return item information with field categories (ex: "Author: James Smith") if item is found, if not, inform user no inventory is found and request if adding such item in the inventory is in need.

ISCat

ISCat is an enum and contains a constructor that declares individual fields of the Book, Cd, and Dvd class, as well as accessor methods for ISModel (the Model Component) to access that field info and pass along to ISView (the View Component).

ISModel

ISModel gets categorized user inputs from ISController. Based on the input received, Model determines the category of the item (book/CD/DVD), and its corresponding fields, and the type of action (add/search/delete/edit) the user wants to perform. After ISModel makes such determinations, ISView calls method PropsActions () to perform corresponding action.

ISView

ISView (the View Component) makes use of the Scanner class to pass through user/keyboard inputs to the system. The item selection method guides user through the different information s/he needs to

provide (Item Categories, Actions needed to be performed, the field number for which the action will be performed if necessary).

ISView calls controller to accept user requests by calling ISController methods to pass user inputs to ISModel through ISModel.

After user inputs has been pass to ISModel, ISModel will provide info to IsView on user selected category and fields. After receiving such Info, ISView calls ISModel method PropsActions () using the modelInteraction method.

ISController

ISController receives user inputs as ISView calls ISController methods such as set Operation (set Model Operations (setMOps()) methods. After categorizing and interpreting those inputs, ISController pass those inputs to ISModel.