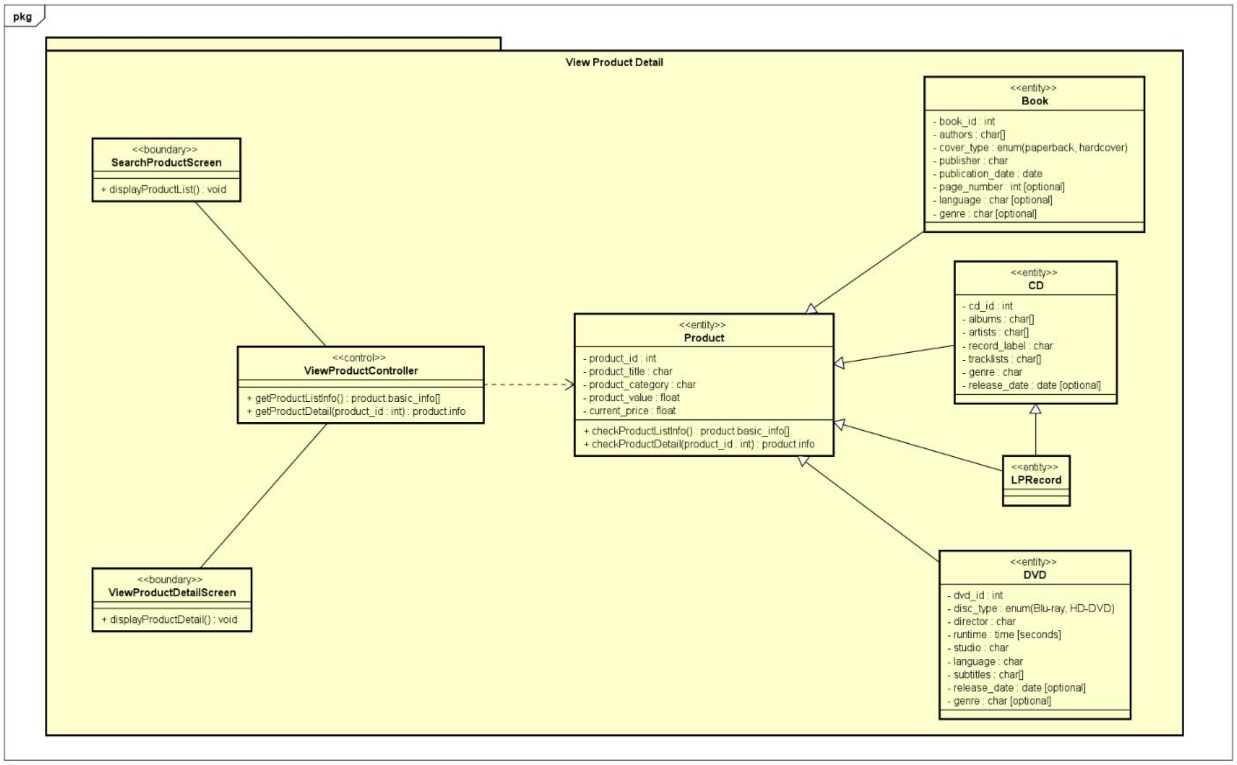
Use Case: View Product Detail

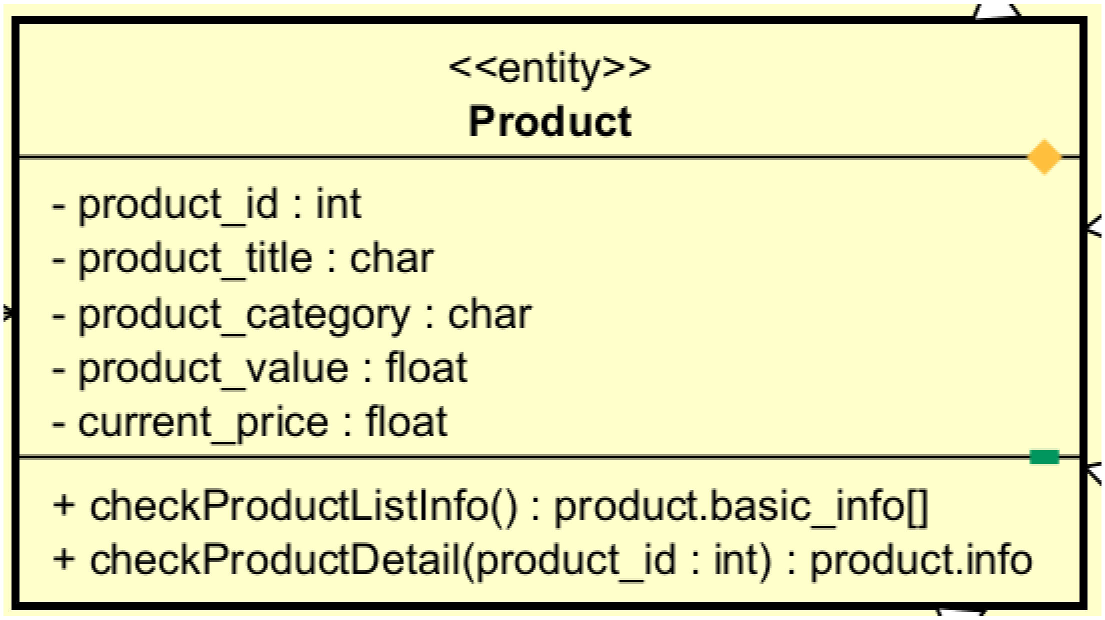
1. Class Design


pkg
View Product Detail
<<boundary>>
SearchProductScreen
+ displayProductList() : void
<<control>>
ViewProductController
+ getProductListInfo() : product.basic_info[]
+ getProductDetail(product_id : int) : product.info
<<boundary>>
ViewProductDetailScreen
+ displayProductDetail() : void
<<entity>>
Product
- product_id : int
- product_title : char
- product_category : char
- product_value : float
- current_price : float
+ checkProductListInfo() : product.basic_info[]
+ checkProductDetail(product_id : int) : product.info
<<entity>>
Book
- book_id : int
- authors : char[]
- cover_type : enum(paperback, hardcover)
- publisher : char
- publication_date : date
- page_number : int [optional]
- language : char [optional]
- genre : char [optional]
<<entity>>
CD
- cd_id : int
- albums : char[]
- artists : char[]
- record_label : char
- tracklists : char[]
- genre : char
- release_date : date [optional]
<<entity>>
LPRecord
<<entity>>
DVD
- dvd_id : int
- disc_type : enum(Blu-ray, HD-DVD)
- director : char
- runtime : time [seconds]
- studio : char
- language : char
- subtitles : char[]
- release_date : date [optional]
- genre : char [optional]

2. Design Class

2.1. Entity

2.1.1. Product Class


<<entity>>
Product
- product_id : int
- product_title : char
- product_category : char
- product_value : float
- current_price : float
+ checkProductListInfo() : product.basic_info[]
+ checkProductDetail(product_id : int) : product.info

Attributes Design:

#	Name	Data type	Default value	Description
1	product_id	Integer	next_id_val	A unique integer used to identifes a product
2	product_title	String		Title of the product
3	product_category	String		Category of the product
4	product_value	Numeric		The value of the product
5	current_price	Numeric		The current price on the website

Operations Design:

#	Name	Return type	Description (purpose)
1	checkProductListInfo()	product.basic_info[]	Returns a list of basic information of products to display (product_title, product_category, current_price)

2	checkProductDetail()	product.info	Returns the detail information of the product with the product_id, including all product-specific information (Ex: albums, artists,… if a product is a CD)

Parameters:

- product_id (checkProductDetail): The product_id of the specific product that is going to get the detailed information.

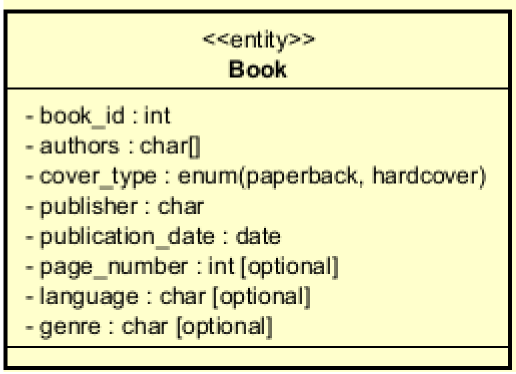
Exceptions:

- invalidProductId: Thrown if 'product_id' does not exist or not a number.

Methods:

- checkProductListInfo()
  Flow:
  - Return a list of product basic information from self.
- checkProductDetail(product_id:int):product.info
  Flow:
  - get the product_category of the product.
  - call the checkProductDetail() operation of the appropriate subclass (Books, CD, LP records, DVD).
  - Merge the general product information with the category-specific information and format it into appropriate response to return to the user.

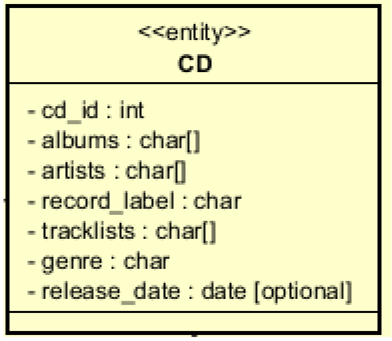
2.1.2. Book Class


<<entity>>
Book
- book_id : int
- authors : char[]
- cover_type : enum(paperback, hardcover)
- publisher : char
- publication_date : date
- page_number : int [optional]
- language : char [optional]
- genre : char [optional]

Attributes Design:

#	Name	Data type	Default value	Description
1	book_id	int	next_id_val	A unique integer used to identifes a book
2	authors	char[]		Name of the book's authors
3	cover_type	enum(paperback, hardcover)		Type of book cover: paperback or hardcover
4	publisher	char		The publisher of the book
5	publication_date	Date		The publication date of the book
6	page_number	int [optional]		Number of page of the book, can be optional
7	language	char [optional]		language of the book, can be optional
8	genre	char [optional]		genre of the book, can be optional

2.1.3. CD Class

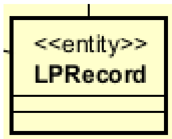


Attributes Design:

#	Name	Data type	Default value	Description
1	cd_id	int	next_id_val	A unique integer used to identifes a CD
2	albums	char[]		Name of the albums inside the CD
3	artists	char[]		Name of the artists working on the CD
4	record_label	char		Name of the record label of the CD

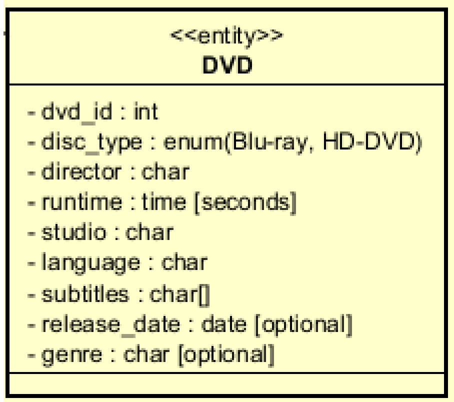
5	tracklists	char[]		Name of the tracklists inside the CD
6	genre	char [optional]		Genre of the CD, can be optional
7	release_date	date [optional]		release date of the CD, can be optional

2.1.4. LPRecord Class



This class inherits all similar attributes of CD class, but instead of cd_id, it has the lp_id attribute.

2.1.5. DVD Class

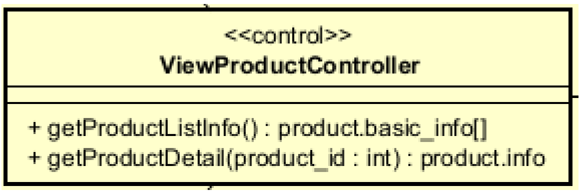

<<entity>>
DVD
- dvd_id : int
- disc_type : enum(Blu-ray, HD-DVD)
- director : char
- runtime : time [seconds]
- studio : char
- language : char
- subtitles : char[]
- release_date : date [optional]
- genre : char [optional]

Attributes Design:

#	Name	Data type	Default value	Description
1	dvd_id	int	next_id_val	A unique integer used to identifes a dvd
2	disc_type	enum(Blu-ray, HD-DVD)		the type of the disc: Blu-ray or HD-DVD
3	director	char		Name of the director of the DVD
4	runtime	time [seconds]		The runtime of the the DVD in seconds
5	studio	char		The studio that create the DVD

6	language	char		The language of the DVD
7	subtitles	char[]		The number of subtitles for the DVD Ex: English, Vietnamese,…
8	release_date	date [optional]		The release date of the DVD, can be optional
9	genre	char [optional]		the genre of the DVD, can be optional

2.2. Controller


<<control>>
ViewProductController
+ getProductListInfo() : product.basic_info[]
+ getProductDetail(product_id : int) : product.info

Operations Design:

#	Name	Return type	Description (purpose)
1	getProductListInfo()	product.basic_info[]	Returns a list of basic information of products to display (product_title, product_category, current_price)
2	getProductDetail()	product.info	Returns the detail information of the product with the product_id, including all product-specific information (Ex: albums, artists,… if a product is a CD)

Parameters:

- product_id (checkProductDetail): The product_id of the specific product that is going to get the detailed information.

Exceptions:

- invalidProductId: Thrown if ‘product_id’ does not exist or not a number.

Methods:

- getProductDetail(product_id:int):product.info
  Flow:
  - Call the checkProductDetail() operation of the Product class and wait for response.
  - Format the repsonse and return it.
- getProductListInfo()
  Flow:
  - Call the getProductListInfo() operation of the Product class and wait for response.
  - Format the repsonse and return it.