Simple software design

Book management

- View book(s)
- Add book
- Modify book
- Delete book

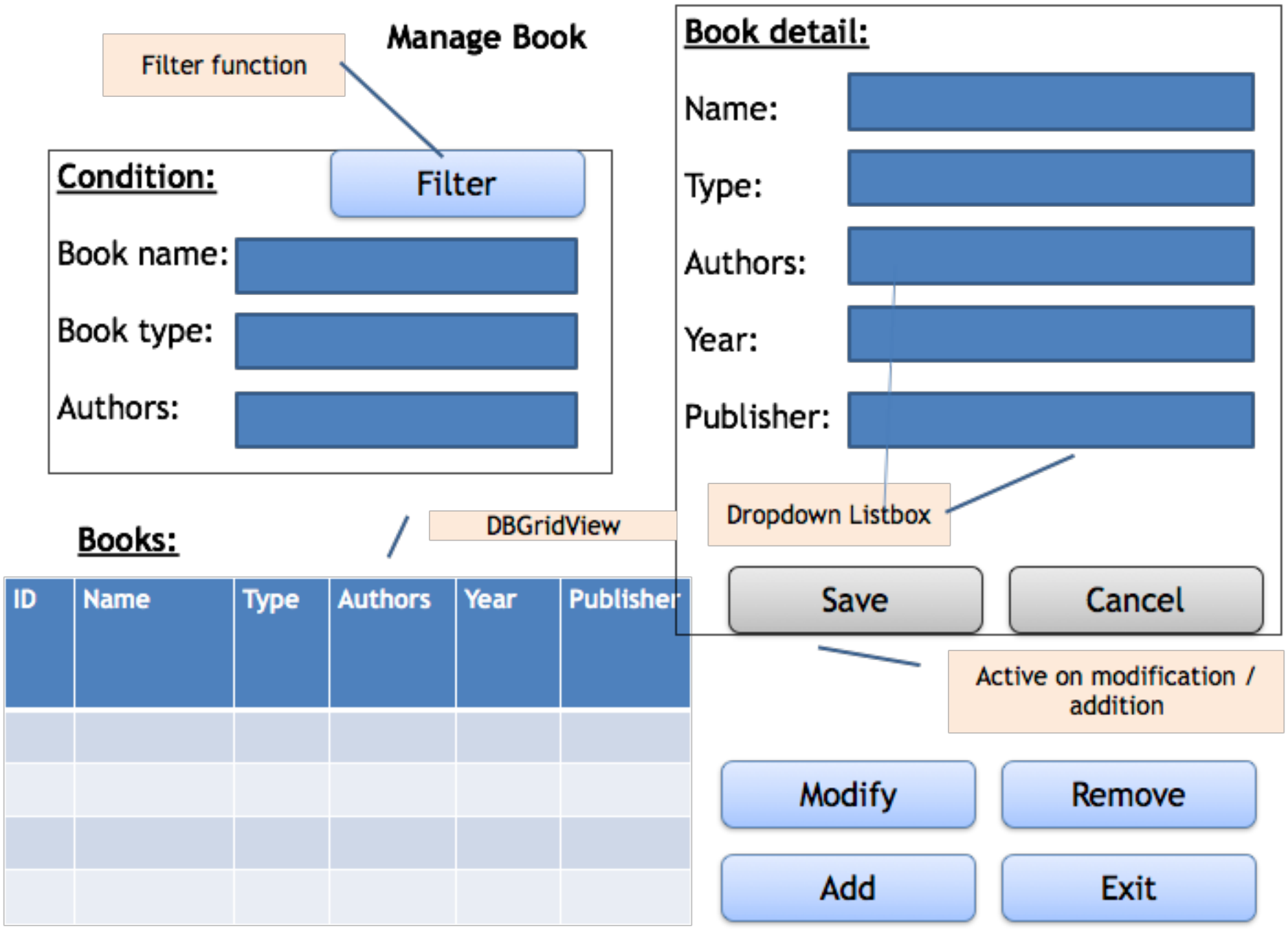
Requirements

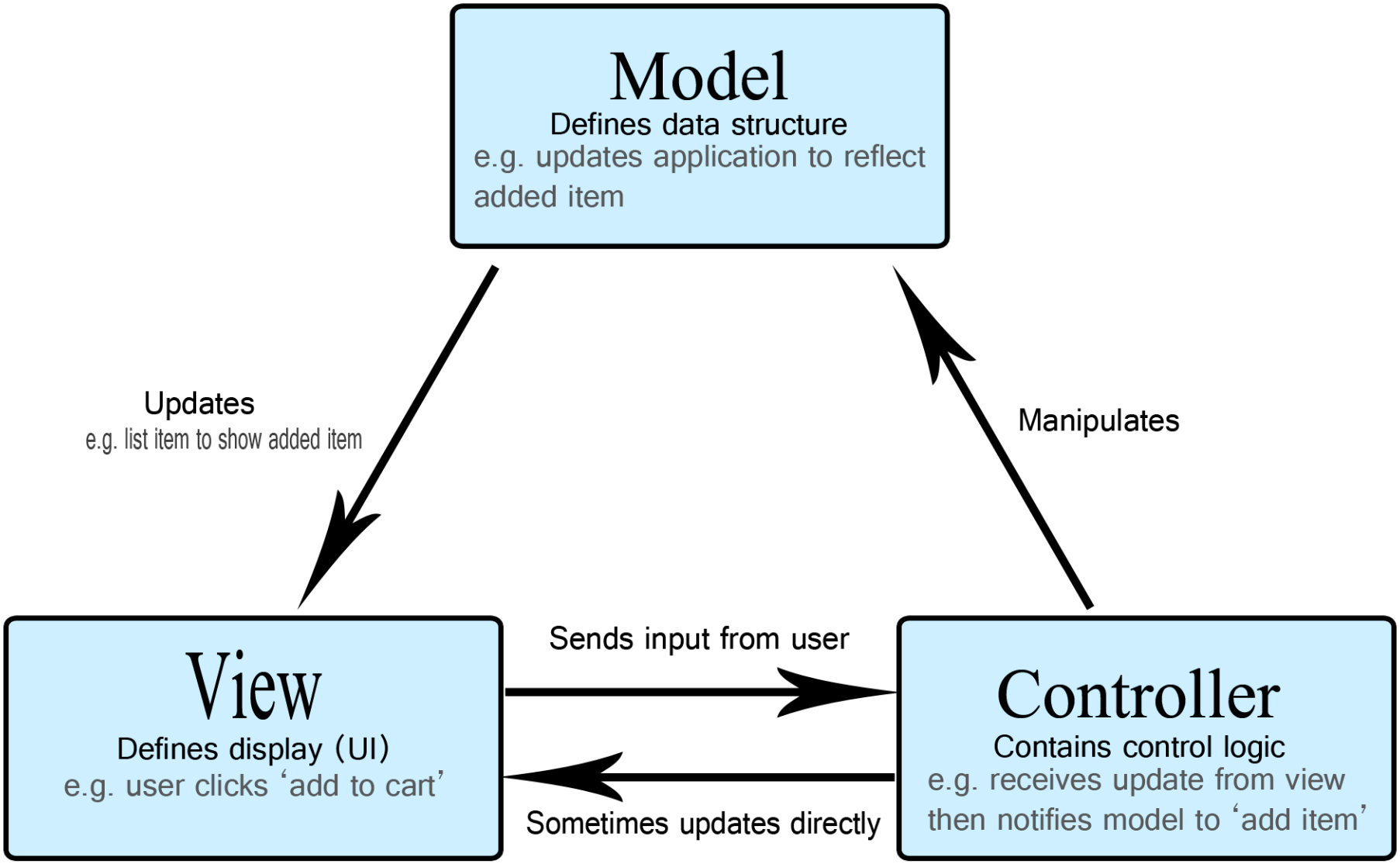
- **Use-case: Manage book information**
- Actor: Book manager
- Stimulus: Book manager chooses “Manage book information” in the main menu
- Main flow:
- 1. System opens the Manage Book window
- 2. System gets the list of books
- 3. System displays the list of books
- 4. Book manager chooses a book from the list
- 5. System gets the book detail and shows
- Extended points: Modify a book. Add a book, Remove a book

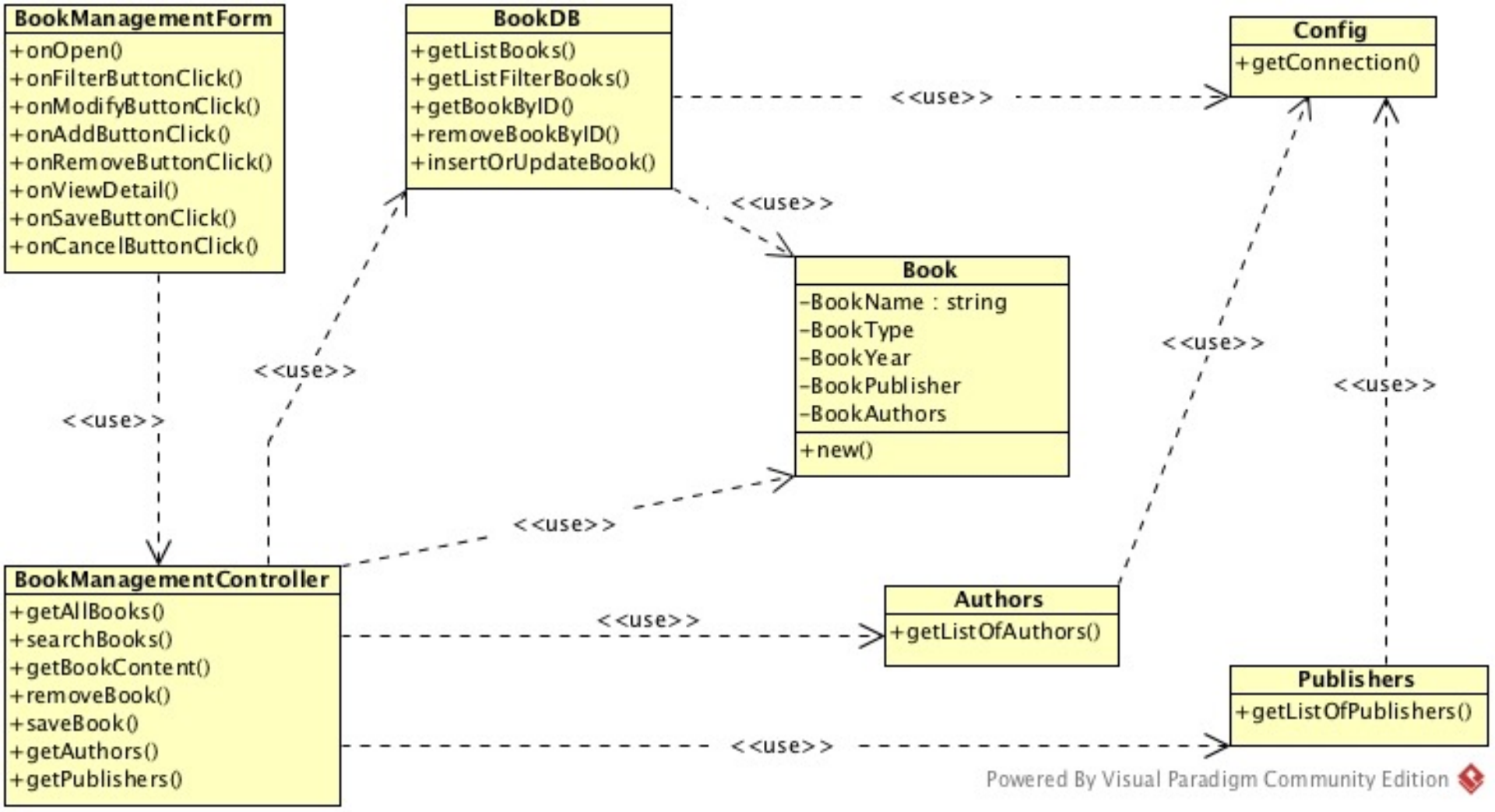
- **Use-case: Remove a book**
- Stimulus: Book manager clicks “Remove” button
- Main flow
- 1. System shows a dialog to confirm the removing
- 2. Book manager confirms
- 3. System remove the book

- **Use-case: Add a book**
- Stimulus: Book manager clicks “Add new” button
- Main flow
- 1. System changes into the edit mode
- 2. Book manager add the detail
- 3. Book manager selects “Save” button
- 4. System saves the new detail

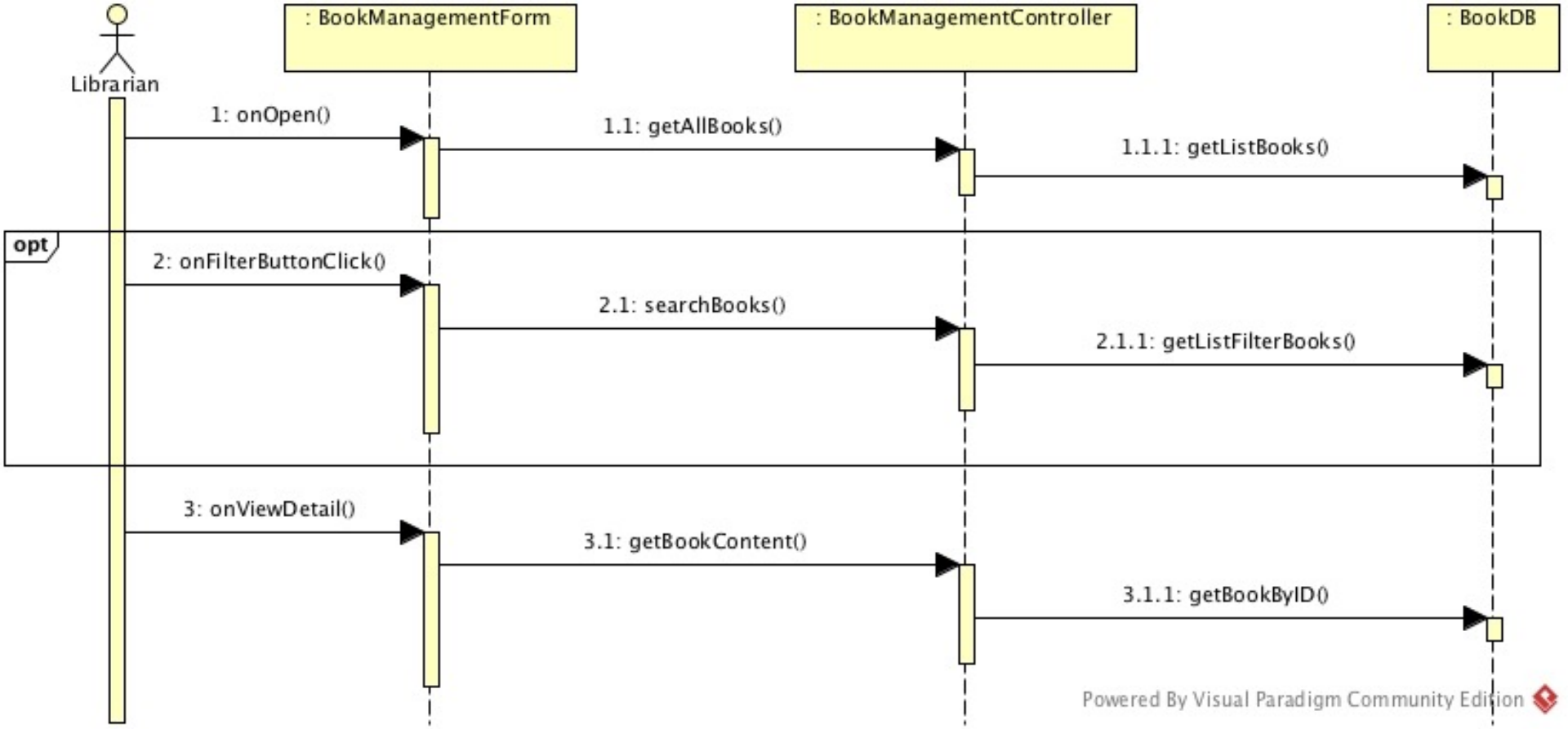
- **Use-case: Modify a book**
- Stimulus: Book manager clicks “Modify” button
- Main flow
- 1. System changes into the edit mode
- 2. Book manager changes the detail
- 3. Book manager selects “Save” button
- 4. System saves the new detail


Manage Book
Filter function
Condition:
Filter
Book name:
Book type:
Authors:
Book detail:
Name:
Type:
Authors:
Year:
Publisher:
Dropdown Listbox
DBGridView
Books:
ID
Name
Type
Authors
Year
Publisher
Save
Cancel
Active on modification / addition
Modify
Remove
Add
Exit


Model
Defines data structure
e.g. updates application to reflect added item
Updates
e.g. list item to show added item
Manipulates
View
Defines display (UI)
e.g. user clicks 'add to cart'
Sends input from user
Sometimes updates directly
Controller
Contains control logic
e.g. receives update from view then notifies model to 'add item'


BookManagementForm
+onOpen()
+onFilterButtonClick()
+onModifyButtonClick()
+onAddButtonClick()
+onRemoveButtonClick()
+onViewDetail()
+onSaveButtonClick()
+onCancelButtonClick()
BookDB
+getListBooks()
+getListFilterBooks()
+getBookByID()
+removeBookByID()
+insertOrUpdateBook()
Config
+getConnection()
Book
-BookName : string
-BookType
-BookYear
-BookPublisher
-BookAuthors
+new()
BookManagementController
+getAllBooks()
+searchBooks()
+getBookContent()
+removeBook()
+saveBook()
+getAuthors()
+getPublishers()
Authors
+getListOfAuthors()
Publishers
+getListOfPublishers()
<<use>>
Powered By Visual Paradigm Community Edition

Sequence diagram for the main use-case



BookManagementForm

- This is a C# form
- onOpen
  - Fired when the form is opened
  - Algorithm:
    - Call BookManagementController.getAllBooks to return a result
    - set (DBGridView) listBook.dataSource = result
    - Disable all controls on Editing frame
- onFilterButtonClick
  - Fired when user clicks on the Filter button
  - Algorithm:
    - Call BookManagementController.searchBook (bookName, bookType, bookAuthors) to return a DataSet result
    - If the DataSet is not empty then set (DBGridView) listBook.dataSource = result
  - Otherwise, set listBook.dataSource = null

BookManagementForm (cont.)

- onViewDetail
  - Fired when user changes the selection on listBook (DBGridView)
  - Algorithm
    - Get the ID value of the selected row of listBook into bookID
    - Call BookManagementController.getBookContent (bookID)
    - Fill the TextBox controls on the form
- onModifyButtonClick
  - Fired when user clicks on the Edit button
  - Algorithm:
    - If there is no data in the TextBox controls, exit
    - Get the list of authors by calling BookManagementController.getAuthors and populate the list box Authors on the form
    - Get the list of publishers by calling BookManagementController.getPublishers and populate the list box Publishers on the form
    - Change the TextBox controls to enable them in editing
    - Enable the Save and Cancel buttons

BookManagementController

- public static ResultSet getAllBooks ()
  - Algorithm:
    - return BookDB.getListBooks ()
- public static ResultSet searchBooks (String bookName, String bookType, String bookAuthors)
  - Algorithm:
    - return BookDB.getListFilterBook (bookName, bookType, bookAuthors)
- public static Book getBookContent (Long bookID)
  - Algorithm:
    - return BookDB.getBookByID (bookID)
- public static boolean removeBook (Long bookID)
  - Algorithm:
    - return BookDB.removeBookByID (bookID)

BookDB

- public static ResultSet getListBooks()
- Algorithm:
  - open a Database Connection using Config.getConnection()
  - generate a SQL statement to select all records from the tbBook joined with tbAuthor and tbPublisher tables
  - execute a Statement on the SQL statement and return the result
- public static ResultSet getListFilterBook (String bookName, string bookType, string bookAuthors)
- Algorithm:
  - open a Database Connection using Config.getConnection()
  - generate a SQL statement to select records from the tbBook joined with tbAuthor and tbPublisher tables with the given conditions
  - execute a Statement on the SQL statement and return the result
- public static Sach getBookByID(Long bookID)
- Algorithm:
  - open a Database Connection using Config.getConnection()
  - generate a SQL statement to select a record from the tbSach joined with tbBook joined with tbAuthor and tbPublisher tables with the given bookID
  - execute a Statement on the SQL statement
  - return a new Book object with the data from the ResultSet if any or return NULL