Problem identification and requirements analysis

Case Study: ReadX Library

Customer	Egyptian conglomerate ReadX
User	ReadX users (regular and premium)
Functional requirements	**FR0:** Register users **FR1:** Register books **FR2:** Register magazines **FR3:** Establish an initial state of the program **FR4:** Modify bibliographic products **FR5:** Deleting bibliographic products **FR6:** Buying books **FR7:** Subscribe to magazines **FR8:** Cancel magazine subscriptions **FR9:** Simulate reading session **FR10:** Present Library of Bibliographic Products **FR11:** Generate reports
Context of the problem	An Egyptian conglomerate is looking for a software prototype that will enable them to manage their publishing business of bibliographic products, which are currently books and magazines. The software must be capable of allowing the administrator to manage all the bibliographic products, in addition to registering both Premium and Regular customer users, so that they can purchase the books and subscribe to the magazines according to their permissions. In addition, the software must be easy to use, user-friendly and its design must allow for easy updates.
Non-functional requirements	**NRF0:** The software should be quick and easy to use for the user, with an easy to understand interface. **NRF1:** The software should be scalable, that is, easy to maintain and update.In addition, it should be designed in such a way that it allows the creation of new types of bibliographic products and users in the future in an easy way.

Identifier and name	*FR0: Register users*
Summary	The software must allow the registration of users (for the moment premium and regular). For this, the system must request and register a

<table>
<tr><td></td><td colspan="3">name and ID for each one. Internally, the software must register the date of linking.

At the time of registration, the user must select the type of account he/she wishes to create, either regular or premium. The system will enable the corresponding functionalities according to the type of account selected.

The software must allow the regular user to: purchase a maximum of 5 books, subscribe to a maximum of 2 magazines. Additionally, the software must present advertisements during their use of the platform.

The software must allow the premium user to: buy unlimited books, subscribe to unlimited magazines.</td></tr>
<tr><td rowspan="4">Input</td><td>Name of Input</td><td>Data type</td><td>Condition valid values</td></tr>
<tr><td>name</td><td>String</td><td>Characters (text), non-empty and obligatory</td></tr>
<tr><td>id</td><td>String</td><td>Characters (text), non-empty and obligatory</td></tr>
<tr><td>userType</td><td>int</td><td>Numbers, no void</td></tr>
<tr><td>Result or Postcondition</td><td colspan="3">The software must perform the following actions:

1. Register the information provided by the user and create the corresponding account, either premium or regular.
2. Issue a message to the user informing whether the registration operation was successful or if an error occurred during the process. In addition, the information of the user who has been registered should be displayed.</td></tr>
<tr><td rowspan="2">Output</td><td>Output name</td><td>Data type</td><td>Format</td></tr>
<tr><td>msg</td><td>String</td><td>Characters (text)</td></tr>
</table>

<table>
<tr><td>Identifier and name</td><td colspan="3">FR1: Register books</td></tr>
<tr><td>Summary</td><td colspan="3">The software must allow the administrator user to register books. To do so, the system must request and record the following information: name, number of pages, short review, a publication date, a genre (Science Fiction, Fantasy and Historical Novel), URL leading to a repository with the book cover and the retail value in dollars. Internally the software must generate a unique identifier of 3 hexadecimal characters.</td></tr>
<tr><td rowspan="2">Input</td><td>Name of Input</td><td>Data type</td><td>Condition valid values</td></tr>
<tr><td>name</td><td>String</td><td>Characters (text), non- empty and obligatory</td></tr>
</table>

	pages	int	*Numbers, no blank and mandatory*
	pubicationDate	Calendar	*Date format:* *mm/dd/yyyy*
	review	String	*Characters (text), non-empty and obligatory*
	genre	enum	*Selection:* *(1. Science Fiction, 2. Fantasy, 3. Historical Novel)*
	url	String	*Characters (text), non-empty and obligatory*
	price	double	*Numbers, no blank and mandatory*
Result or Postcondition	The software must perform the following actions: 1. Record the information supplied by the user and register the book, internally generating a unique identifier for the book. 2. Issue a message to the user informing whether the registration operation has been performed correctly or if an error has occurred during the process. In addition, the information of the book that has been registered should be displayed.		
Output	**Output name**	**Data type**	**Format**
	msg	String	*Characters (text)*

Identifier and name	*FR2: Register magazines*		
Summary	The software must allow the administrator user to register journals. To do so, the system must request and register the following information: name, number of pages, date of publication, category (Varieties, Design and Scientific), URL that leads to a repository with the cover of the journal, subscription value (in dollars) and the periodicity of issue. Internally the software must generate a unique identifier of 3 alphanumeric characters.		
Input	**Name of Input**	**Data type**	**Condition valid values**
	name	String	*Characters (text), non-empty and obligatory*

	pages	int	*Numbers, no blank and mandatory*
	pubicationDate	Calendar	*Date format:* *mm/dd/yyyy*
	category	enum	*Selection:* *(1. Varieties, 2. Design, and 3. Scientific)*
	url	String	*Characters (text), non-empty and obligatory*
	price	double	*Numbers, no blank and mandatory*
	periodicity	String	*Characters (text), non-empty and obligatory*
Result or Postcondition	The software must perform the following actions: 1. Register the information provided by the user and register the journal, internally generating a unique identifier for the journal. 2. Issue a message to the user informing if the registration operation has been performed correctly or if an error has occurred during the process. In addition, the information of the journal that has been registered should be displayed.		
Output	**Output name**	**Data type**	**Format**
	msg	String	*Characters (text)*

Identifier and name	*FR3: Establish an initial state of the program*		
Summary	The software must establish an initial state of the program in which bibliographic products and users, both regular and premium with their corresponding data, exist to allow the proper use of the system. To this end, the system should have an initialization process that creates a set of bibliographic products and test users.		
Input	**Name of Input**	**Data type**	**Condition valid values**
	N/A	N/A	N/A
Result or Postcondition	The software must perform the following actions: 1. Create 1 regular user 2. Create 1 premium user		

	3. Create 1 magazine for each category. 4. Create 1 book for each genre.		
Output	**Output name**	**Data type**	**Format**
	N/A	N/A	N/A

Identifier and name	*FR4: Modify bibliographic products*		
Summary	The software must allow the administrator user to modify existing bibliographic products. To do this, the system should request the product ID and locate the product. Once the product is located, the user will be able to choose which data to modify and provide its new status. The system should check the validity of the new value provided by the user before updating the bibliographic product. If the new value is not valid, the system should inform the user and request a valid Input.		
Input	**Name of Input**	**Data type**	**Condition valid values**
	productId	String	*Must be the id of an existing product*
	dataToModify	int	*Numbers, no vacuum*
	newStatus	String	*It must correspond to the type of data you wish to modify and must not be empty.*
Result or Postcondition	The software must perform the following actions: 1. Change the status of the data to be modified with the new value specified by the user. 2. Issue a message to the user informing whether the modification operation has been performed correctly or if an error has occurred during the process. In addition, the updated information of the bibliographic product that has been modified should be displayed.		
Output	**Output name**	**Data type**	**Format**
	msg	String	*Characters (text)*

Identifier and name	*FR5: Deleting bibliographic products*
Summary	The software must allow the administrator user to delete existing bibliographic products. To do so, the system must request the id of the

<table>
<tr><td></td><td colspan="3">product to be deleted and locate it. Once the product is located, the software should delete it from the system.</td></tr>
<tr><td rowspan="2">Input</td><td>Name of Input</td><td>Data type</td><td>Condition valid values</td></tr>
<tr><td>productId</td><td>String</td><td>Must be the id of an existing product</td></tr>
<tr><td>Result or Postcondition</td><td colspan="3">The software must perform the following actions:
1. If the product exists, delete it and issue a message to the user indicating that the operation was successful.
2. If the product does not exist, issue a message to the user indicating that the product has not been found in the system and deletion cannot be performed.</td></tr>
<tr><td rowspan="2">Output</td><td>Output name</td><td>Data type</td><td>Format</td></tr>
<tr><td>msg</td><td>String</td><td>Characters (text)</td></tr>
</table>

<table>
<tr><td>Identifier and name</td><td colspan="3">FR6: Buying books</td></tr>
<tr><td>Summary</td><td colspan="3">The software must allow both regular and premium users to purchase books according to their permissions. To do this, the system must request the name of the book and locate it. Once the book is located, the system should display its information and ask the user if he/she wants to buy it. If the book is not found in the system, the system must inform the user.

When the user purchases the book, the system should generate an invoice containing the date of the transaction and the amount paid. In addition, the system should automatically update the number of copies of the book sold.

In case the user does not have permissions to buy a book, the system must inform the user.</td></tr>
<tr><td rowspan="2">Input</td><td>Name of Input</td><td>Data type</td><td>Condition valid values</td></tr>
<tr><td>bookName</td><td>String</td><td>Must be the name of an existing product</td></tr>
<tr><td>Result or Postcondition</td><td colspan="3">The software must perform the following actions:
1. Add the purchased book to the user's book list and update the number of copies sold.
2. Generate and display the purchase invoice with the date of the transaction and the amount paid.
3. Issue a message to the user informing if the purchase operation has been carried out correctly or if an error has occurred during the process.</td></tr>
</table>

Output	**Output name**	**Data type**	**Format**
	msg	String	*Characters (text)*

Identifier and name	*FR7: Subscribe to magazines*		
Summary	The software must allow both regular and premium users to subscribe to magazines according to their permissions. To do this, the system must request the name of the magazine and locate it. Once the magazine is located, the system will display its information and ask the user if he/she wants to purchase it. If the magazine is not found in the system, the system must inform the user. When the user subscribes to a magazine, the system must generate an invoice containing the date of the operation and the amount paid. In addition, the system must automatically update the number of active subscriptions to the magazine. In case the user does not have permissions to subscribe to a magazine, the system must inform the user.		
Input	**Name of Input**	**Data type**	**Condition valid values**
	magazineName	String	*Must be the name of an existing product*
Result or Postcondition	The software must perform the following actions: 1. Add the magazine to the user's magazine list and update the number of active subscriptions. 2. Generate and display the purchase invoice with the date of the transaction and the amount paid. 3. Issue a message to the user informing if the purchase operation has been carried out correctly or if an error has occurred during the process.		
Output	**Output name**	**Data type**	**Format**
	msg	String	*Characters (text)*

Identifier and name	*FR8: Cancel magazine subscriptions*
Summary	The software should allow both regular and premium users to cancel their magazine subscriptions. To do this, the system should request the name of the magazine and locate it in the user's list of magazines. Once the magazine is located, the system will display its information, confirm with the user whether to cancel the subscription and remove it

<table>
<tr><td></td><td colspan="3">from the list if applicable. If the journal is not found, the system must inform the user.</td></tr>
<tr><td rowspan="2">Input</td><td>Name of Input</td><td>Data type</td><td>Condition valid values</td></tr>
<tr><td>magazineName</td><td>String</td><td>Must be the name of an existing product</td></tr>
<tr><td>Result or Postcondition</td><td colspan="3">The software must perform the following actions:
1. Remove the magazine from the user's subscription list.
2. Issue a message to the user informing if the purchase operation has been carried out correctly or if an error has occurred during the process.</td></tr>
<tr><td rowspan="2">Output</td><td>Output name</td><td>Data type</td><td>Format</td></tr>
<tr><td>msg</td><td>String</td><td>Characters (text)</td></tr>
</table>

<table>
<tr><td>Identifier and name</td><td colspan="3">FR9: Simulate reading session</td></tr>
<tr><td>Summary</td><td colspan="3">The software must allow both regular and premium users to read their bibliographic products. To do this, the system should request the user's identifier and product, locate it and display on screen the name of the bibliographic product, the current page being read and navigation options to read the previous page “A”, the next page “S” and to return to the Library “B”.

Each time the user changes pages the system should automatically update the number of pages read for that product and save the user's progress so that he/she can resume reading at the point where he/she left off.

The software should present advertisements to the regular user when starting a reading and after every 20 pages read of a book or 5 pages read of a magazine.

The advertisements will appear randomly among the following:
• Subscribe to Combo Plus and get Disney+ and Star+ at an incredible price!
• Now your pets have a favorite app: Laika. The best products for your furry friend.
• It's our anniversary! Visit your nearest Éxito and be surprised with the best offers.</td></tr>
<tr><td rowspan="2">Input</td><td>Name of Input</td><td>Data type</td><td>Condition valid values</td></tr>
<tr><td>input</td><td>String</td><td>May be:
• A
• S</td></tr>
</table>

			• *B*
Result or Postcondition	The software must perform the following actions: 1. Show the user the pages he/she wants to read and update the number of pages read. 2. Present advertisements periodically to the regular user. 3. Allow the user to return to the Library at any time during reading.		
Output	**Output name**	**Data type**	**Format**
	currentPage	String	*Characters (text)*

Identifier and name	*FR10: Present Library of Bibliographic Products*		
Summary	The software should be able to present each regular and premium user with a library of his or her bibliographic products. To do this, the system should request the user's identification number, locate his library and present it. This library consists of 5x5 shelves (matrices) containing the identification numbers of the bibliographic products held by the user, the products should be organized by publication date, from the oldest to the newest. As the user acquires more bibliographic products, new shelves should be generated. The system should allow the user to browse the shelves of his or her library. If the user enters the letter "A", the user should be taken back to the previous shelf. If the letter "S" is entered, the user should advance to the next shelf. If the letter "E" is entered, the user must exit the library and return to the main menu. In addition, the system should allow the user to select a specific bibliographic product to initiate a reading session. This can be done by its identification number or by the coordinates (x, y) of the current shelf.		
Input	**Name of Input**	**Data type**	**Condition valid values**
	Input	String	*May be:* • *A* • *S* • *E*
	userId	String	*Must be the id of an existing user*
	productId	String	*Must be the id of an existing product*

Result or Postcondition	The software must perform the following actions: 1. Start the reading session of the bibliographic product selected by the user.		
Output	**Output name**	**Data type**	**Format**
	library	String	*Characters (text)*

Identifier and name	*FR11: Generate reports*		
Summary	The software must allow the generation of the following reports: • For each type of bibliographic product, book and journal, report the total accumulated pages read across the platform (type of product and number of pages read). • Report the most read genre or category across the platform, this report should contain the name of the genre or category and number of pages read. • Top 5 most read books and magazines report: This report will present the five most read books and five most read magazines on the platform, including the name of the book or magazine, its respective genre or category, and the number of pages read. • Sales report by book genre: For each book genre, this report will report the number of books sold and the total dollar value of sales. • Active subscriptions report and total value paid by magazine category: For each magazine category, this report will show the number of active subscriptions and the total value paid for those subscriptions.		
Input	**Name of Input**	**Data type**	**Condition valid values**
	N/A	N/A	*N/A*
Result or Postcondition	The software must present the different reports.		
Output	**Output name**	**Data type**	**Format**
	report	String	*Characters (text)*