

TABLA DE ESPECIFICACIÓN DEL PROBLEMA DE INGENIERÍA DE SOFTWARE, identificando los siguientes elementos

CUSTOMER	ReadX
USER	ReadX manager, Reader
FUNCTIONAL REQUERIMENTS	<ul style="list-style-type: none"> • Manage users, regular and premium • Manage bibliographic products • Generate objects automatically • Consult bibliographic products by name • Allow a user to buy a book • Allow a user to subscribe to a magazine • Allow a user to unsubscribe from a magazine • Present your library of bibliographic products to the user • Allow a user to simulate a read session • report the accumulated total number of pages read for each type of bibliographic product • Report the most read book genre and magazine category on the platform • Report the Top 5 most read books and Top 5 most read magazines on the platform • For each gender, report the number of books sold and the total value of sales • For each category, report the number of active subscriptions and the total value paid for subscriptions
CONTEXT	<p>ReadX needs software to run a global publishing business. Where they need a manager to manage their bibliographic products and users. The software must allow buying books or subscribing to magazines for the different types of readers (premium ones who should not see ads and regular ones who do have ads), there must also be a presentation of each reader's library showing a matrix of 5x5 all the contents bibliographical products that you have registered. In addition, it will be possible to simulate readings by going through the pages between your products showing ads if your type of user requires it. The software must also be able to create different reports and tests so that the</p>

	manager can be aware of the software data and its content.
NON-FUNCTIONAL REQUERIMENTS	<ul style="list-style-type: none"> • Scalability: The system must be able to handle a large number of bibliographic products, as it is expected that many will be added over time. • Performance: The system must be able to handle multiple user requests at the same time (subscribe, view library, etc.), without long waiting times. • Usability: The system must be easy to use and intuitive for end users, with a clear and well-designed interface that allows easy navigation to purchase, subscribe, read, and report. • Integration: The system must be compatible with other company tools to guarantee greater efficiency and reduce the duplication of information such as information from readers, managers and the content of bibliographic products. • Maintenance: The system must be easy to maintain and update, with clear and complete documentation for the personnel in charge of its administration and maintenance of the book platform. • Traceability: The system must allow the traceability of bibliographic products, from their registration to their purchases or subscriptions, to guarantee greater transparency in the process and better ReadX reports.

Tabla de análisis de requerimientos funcionales (Nota: Una tabla por cada requerimiento funcional)

Name or id	FR1: Manage users, regular and premium		
Abstract	The System asks the user to register a reader through its name, id and type (regular and premium). Then the system gets the current registration date and adds the user to the databases.		
Inputs	Input name	Datatype	Selection Condition
	nameReader	String	
	idReader	String	Numbers
	String	typeReader	1. Premium 2. Regular
Result	The system adds the Reader to the databases, saving your information along with the current date of registration		
Output	Input name	Datatype	Selection Condition
	confirmationRegister	String	"The Reader has been created"

Name or id	FR2: Manage bibliographic products		
Abstract	<p><i>The system allows the manager to manage bibliographic products, the system allows 3 options.</i></p> <p><i>Bibliographic product creation: The manager enters the information of the new bibliographic product depending on whether he wants to enter a book or magazine. The data requested for the book are: name, number of pages, review, publication date, genre, URL that leads to a repository with the book cover, and sale value. The data requested for the magazine are: name, number of pages, date of publication, category, URL that leads to a repository with the cover of the magazine, subscription value and frequency of issue. The system creates the product, creates the new product identifier, and prints your confirmation.</i></p> <p><i>Bibliographic product edition: If the manager wishes to edit some information of a bibliographic product, he can enter its identifier and add the change by varying some attribute of the products. The system edits the product and prints your confirmation.</i></p> <p><i>Elimination of bibliographic product: If the manager wishes to delete a product, he can enter its identifier, the system deletes it from its databases and from the libraries of other readers.</i></p>		
Inputs	Input name	Datatype	Selection Condition
	typeProduct	int	1. Book 2. Magazine
	idProduct	String	for Book:3 hex characters for Magazine: 3 alphanumeric characters
	name	String	
	numberPages	int	numberPages > 0
	publicationDate	Calendar	dd-MM-yyyy
	review	String	
	gender	GenderType	Science Fiction, Fantasy, Historical Novel
	category	CategoryType	Variety, Design, Scientist
	url	String	URL
	valueProduct	double	\$>0
	periodicity	String	

Result	<p><i>The system depending on what is requested by the manager will create a result.</i></p> <p><i>Bibliographic product creation: The system adds the information to the database, creates the product and prints the creation confirmation.</i></p> <p><i>Bibliographic product edition: The system will modify the product information, save the modification and print the edition confirmation.</i></p> <p><i>Delete bibliographic product: The system deletes the information from the database, deletes the product and prints the deletion confirmation.</i></p>		
Output	Input name	Datatype	Selection Condition
	confirmationRegister	String	"The product has been created"
	idProduct	String	3 hex characters
	confirmationEdit	String	"The product has been edited"
	confirmationDelete	String	"The product has been deleted"

Name or id	FR3: Generate objects automatically		
Abstract	The system allows the creation of some base objects to have an easier handling of the different operations of the manager. The system creates regular users, Premium, predetermined books and magazines, allowing the manager to review reports or reading simulations more efficiently.		
Inputs	Input name	Datatype	Selection Condition
Result	The system creates regular, premium, book and magazine users with default attributes for the manager's ease.		
Output	Input name	Datatype	Selection Condition
	confirmationCreation	String	"The objects have been created"

Name or id	FR4: Consult bibliographic products by name		
Abstract	The system asks the user for the name of a product. The system then displays a list of search matches with their codes.		
Inputs	Input name	Datatype	Selection Condition
	searchNameString	String	
Result	The system prints a series of coincidences with the user's search so that the user keeps in mind the codes of the search products.		
Output	Input name	Datatype	Selection Condition
	listProducts	String	List with names and links

Name or id	FR5: Allow a user to buy a book		
Abstract	The system asks the reader to enter their code and that of the book to buy it, the invoice is generated with the date of the operation and the amount to be paid. The system updates the number of copies sold and prints the confirmation. It must be taken into account that if the reader is regular, they cannot go beyond 5 books.		
Inputs	Input name	Datatype	Selection Condition
	idReader	String	number
	idBook	String	3 hex characters
Result	The system registers the purchase of the reader if it has the space, the invoice is printed with the data of the date and the amount. Then, the number of copies sold is added.		
Output	Input name	Datatype	Selection Condition
	amountPayable	Int	valueProduct
	purchaseDate	Calendar	dd-MM-yyyy
	confirmationBuy	String	"The book has been boughth"

Name or id	FR6: Allow a user to subscribe to a magazine		
Abstract	The system asks the reader to enter their code and that of the magazine to subscribe, the invoice is generated with the date of the operation and the amount to be paid. The system updates the number of active subscriptions and prints the confirmation. It must be taken into account that if the reader is regular, they cannot go beyond 2 magazines.		
Inputs	Input name	Datatype	Selection Condition
	idReader	String	numbers
	idMagazine	String	3 alphanumeric characters
Result	The system registers the purchase of the reader if it has the space, the invoice is printed with the data of the date and the amount. Then the number of active subscriptions is added up.		
Output	Input name	Datatype	Selection Condition
	amountPayable	Int	valueProduct
	purchaseDate	Calendar	dd-MM-yyyy
	confirmationSuscribe	String	"You subscribe to the magazine"

Name or id	FR7: Allow a user to unsubscribe from a magazine		
Abstract	The system asks the reader to enter their code and that of the magazine to cancel the subscription. The system removes the subscription and decreases the number of active subscriptions. Then, print the cancellation confirmation.		
Inputs	Input name	Datatype	Selection Condition
	idReader	String	numbers
	idMagazine	String	3 alphanumeric characters
Result	The system unsubscribes the user, decreases active subscriptions, and prints the execution confirmation.		
Output	Input name	Datatype	Selection Condition
	confirmationNotSuscribe	String	"You are not subscribe to the magazine now"

Name or id	FR8: Present your library of bibliographic products to the user		
Abstract	The system asks the reader to enter their identification to be able to print their collection of bibliographic products on the screen in a 5x5 matrix showing their code. These are ordered by publication date (Old -> New), the user can also navigate between the pages he has in his library. The reader can enter the coordinate (x,y) or the 3-character code to start a reading session		
Inputs	Input name	Datatype	Selection Condition
	idReader	String	numbers
	option	String	A - previous page S - next page E - exit
	coordinates	String	X,Y
	idProduct	String	for Book:3 hex characters for Magazine: 3 alphanumeric characters
Result	The system prints the library in the 5x5 matrix, the reader changes the page until he chooses a bibliographic product to start his reading session.		
Output	Input name	Datatype	Selection Condition
	library	String	Matrix 5x5

Name or id	FR9: Allow a user to simulate a read session		
Abstract	When the product is chosen in the library, the system starts a reading session simulation, where the name of the bibliographic product, the current page and the options of next page, previous page and return to the library are displayed. As the reader turns the page, the number of pages read in the bibliographic product increases. In case of being a regular user, ads are presented when the reading session begins and every 20 pages read in a book or 5 pages in a magazine.		
Inputs	Input name	Datatype	Selection Condition
	libraryOption	String	A - previous page S - next page B – Back to library
Result	The user simulates the reading session as long as "return to the library" has not been entered, and the reading simulation and the announcements corresponding to regular users are printed on the screen. The number of pages read for the bibliographic product is increased according to the number of pages read by the session carried out.		
Output	Input name	Datatype	Selection Condition
	page	String	name of the bibliographic product, the current page and the options to do
	ad	AdType	Disney+ y Star+ Laika Éxito