**Design and analysis of the integrative task 3**

**Oscar Stiven Munoz Ramirez**

**A00399922**

**ICESI University**

**Faculty of engineering**

**Engineer Nicolas Salazar**

**Cali, Colombia**

**April 24, 2023**

# Problem identification and requirements analysis

# Case Study : Virtual bookstore program for the Egyptian conglomerate ReadX

|  |  |
| --- | --- |
| Customers | Egyptian conglomerate ReadX |
| Usuario | ReadX program users (regular or premium) |
| Requerimientos funcionales | FR0. Register Users  FR1. Register Bibliographic Product  FR2. Modify Bibliographic Product  FR3. Delete Bibliographic Product  FR4. Program Starter Text  FR5. Allow user to purchase bibliographic products  FR6. Allow User to Cancel Subscription  FR7. Show product bibliographic of the user  FR8. Simulate Reading of Bibliographic Product  FR9. Report Generation |
| Contexto del problema | The problem that arises is the development of a software prototype for the management of the worldwide publishing business of the company ReadX, with the aim of facilitating the management of bibliographic products, users and sales of books and magazines. The prototype must allow the registration, modification and elimination of bibliographic products, as well as the acquisition of books and subscription to magazines by users, and the graphic presentation of the library of products. In addition, the simulation of a reading session must be allowed, with navigation options to go forward or backward in the product pages. All of this must be thought of in a flexible way to accommodate future changes in the publishing industry and the growth of the ReadX platform. |
| Requerimientos no funcionales | FNR0-Code generation  FNR1-Fault Controller  FNR2-User Backups for Purchased Bibliographic Products  Security: The system must be secure to avoid possible attacks or vulnerabilities.  FNR3- Scalability: The system must be able to handle large volumes of data and users without reducing its performance.  FNR4- Usability: The system must be easy to use and intuitive for end users.  FNR5- Availability: The system must be available at all times for users and must not have long downtimes.  FNR6- Usability: The system must be easy to use and intuitive for end users. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identification and name | FR0. Register Users | | | |
| Summary | The system must allow the user to register their account by requesting their name and ID number (the system will instantiate the date of the account created automatically). Additionally, it must be distinguished that there are two types of accounts: premium and regular.  Regular users will be able to buy 5 books and subscribe to up to two periodical publications (magazines), while premium users have unlimited capabilities.  The user IDs cannot be repeated. | | | |
| Input | **Input name** | **Data type** | | **Valid values and conditions** |
| nameUser | String | | Up to 30 characters are allowed, with only letters and spaces. |
| CC | String | | It must allow up to 10 characters, with only numbers and no spaces.. |
| TypeAccount | int | | The system will only validate two types of input:  - 1 (Regular User)  - 2 (Premium User) |
| Result and postcondition | The software will check the data to ensure that the same type of account is not repeated, with the aim of storing the user’s data in the database. Additionally, it will return information on whether the operation was successful or not. | | | |
| Output | **Output name** | | **Data type** | **Format** |
| msg | | String | A text message will be displayed to inform whether the registration was successful or not. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Indetification and name | **FR1. Register Bibliographic products** | | | |
| Summary | The software allows the registration of two types of works: magazines and books, however, these two products share the same characteristics that can be used for inheritance. The data they share are: a unique identifier (3 hexadecimal characters generated automatically), a name, a number of pages, publication date, a URL that leads to a repository of the book and the value of the acquisition of the bibliographic product. However, there will be a flag variable to identify the type of work to be registered, so if the user registers a book, the following data will be requested:  - Choose the type of book: Science Fiction, Fantasy, or Historical Novel  On the other hand, if the user registers a magazine, the following data will be requested:  - Frequency of publication  - Type of magazine category: Variety, Design, and Scientific. | | | |
| Input | **Input name** | **DateType** | | **Valid values conditions** |
| chooseProduct | Int | | Only two options will be allowed:   1. Books 2. 2. Magazine |
| nameProduct | String | | Characters with spaces are allowed. The book name cannot be repeated due to copyright restrictions. |
| AmountPag | int | | Up to 13,228 characters are allowed, as per the Guinness World Records. |
| DatePublication | Calendar | | All types of dates are allowed. If the publication date is before the registration date, it will still be accepted. |
| Url | String | | It must have a protocol, main domain, and an access route (slug). |
| ProductValue | double | | Only real numbers are accepted. |
| TypeBook | int | | Only one of these options can be chosen:  1. Science Fiction  2. Fantasy  3. Historical Novel |
| TypeMagazine | int | | Only one of these options can be chosen:  1. Variety  2. Design  3. Scientific |
| emission | int | | Only these options are allowed:  1. Monthly  2. Quarterly  3. Biannual  4. Annual |
| Result and postcondition | The system will check for any errors, and if everything is okay, it will update the database with the new record. It will also notify the user with a message whether the function worked correctly or not. | | | |
| Output | **Output name** | | **DateType** | **Format** |
| msg | | String | A text message will be displayed to inform the user whether the registration was successful or not. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Indetification and name | **FR2. Modify Bibliographic products** | | | |
| Summary | The software allows to modify two types of works, magazines and books, however, these two products share the same characteristics. Depending on the instance, the program will verify which type of product it is (book or magazine), and according to this, a control panel will appear specifying which properties can be modified (array of integers). Depending on the selected options, the corresponding properties will be available for modification.  The data of the products can be modified, and since books and magazines share the same characteristics, they will share variables. The variables they share are: a unique identifier (3 hexadecimal characters generated automatically), a name, a number of pages, publication date, a URL that leads to a repository of the book or magazine, and the value of the acquisition of the bibliographic product. However, according to the instance of the product, specific information can be modified. If it is a book, the following data will be requested:  - Type of book: Science Fiction, Fantasy, or Historical Novel.  On the other hand, if the user registers a magazine, the following data will be requested:  - Frequency of publication.  - Category of the magazine: Varied, Design, or Scientific.  The software will verify if there are any errors in the modification process and if everything goes well, it will update the database with the new record. A message will be displayed to the user indicating whether the function was carried out correctly or not. | | | |
| Input | **Input name** | **DateType** | | **Valid values conditions** |
| chooseProduct | Int | | Only two options will be allowed:   1. Books 2. 2. Magazine |
| nameProduct | String | | Characters with spaces are allowed. The book name cannot be repeated due to copyright restrictions. |
| AmountPag | int | | Up to 13,228 characters are allowed, as per the Guinness World Records. |
| DatePublication | Calendar | | All types of dates are allowed. If the publication date is before the registration date, it will still be accepted. |
| Url | String | | It must have a protocol, main domain, and an access route (slug). |
| ProductValue | double | | Only real numbers are accepted. |
| TypeBook | int | | Only one of these options can be chosen:  1. Science Fiction  2. Fantasy   1. 3. Historical Novel |
| TypeMagazine | int | | Only one of these options can be chosen:  1. Variety  2. Design   1. 3. Scientific |
| emission | int | | Only these options are allowed:  1. Monthly  2. Quarterly  3. Biannual   1. 4. Annual |
| Result and postcondition | The software will review the data for errors and, if there are none, will save the user’s data in the database. It will also return information on whether the operation was successful or not. | | | |
| Output | **Output name** | | **DateType** | **Format** |
| msg | | String | The system will display a text message informing the user whether the function was carried out successfully or not. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Indetification and name | **FR3. Eliminate Bibliographic products** | | | |
| Summary | The software allows the user to delete bibliographic products, so they only need to enter the name or hexadecimal code of this work. | | | |
| Input | **Input name** | **DateType** | | **Valid values conditions** |
| identification | String | | Maximum 100 characters with spaces are accepted |
| Result and postcondition | The software will verify the data and search for the bibliographic product to delete it. Then, it will return information on whether the operation was successful or not. | | | |
| Output | **Output name** | | **DateType** | **Format** |
| msg | | String | A text string message will be displayed, informing if the function was carried out correctly or not. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Indetification and name | **FR4. Test init** | | | |
| Summary | The software will create a textInit automatically in it, in order to simplify the tests for the end user, it is important that the prototype has a function that allows the automatic generation of at least one object for each type of user and bibliographic product in the system, when the user requires it. | | | |
| Input | **Input name** | **DateType** | | **Valid values conditions** |
|  |  | |  |
| Result and postcondition | The software will take care of creating the objects of this functionality in the database. And it will show a message with the basic statistics of the objects. | | | |
| Output | **Output name** | | **DateType** | **Format** |
| msg | | String | A text string message will be displayed, reporting the data of the created objects. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Indetification and name | **FR5. Allow user to purchase bibliographic products** | | | |
| Summary | The system must allow the user to purchase any bibliographic product. First, you must enter the name or hexadecimal code of the bibliographic product. When you find it. The purchase date will be instantiated automatically and the date of the amount to be paid for the subscription or sale of the product must be entered. | | | |
| Input | **Input name** | **DateType** | | **Valid values conditions** |
| identification | String | | Maximum 100 characters with spaces are accepted |
| valueProducto | double | | You will only be allowed to enter a real number, without spaces. |
| Result and postcondition | The system will evaluate the user’s request and will save the information in the database and will associate the book with the one purchased. However, if the user is regular, it will not allow further linking to his account. | | | |
| Output | **Output name** | | **DateType** | **Format** |
| msg | | String | A text string message will be displayed, informing if the purchase was carried out correctly or not. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Indetification and name | **FR6. Unsubscribe from the magazine to the user** | | | |
| Summary | The software allows the user to unsubscribe from a journal by entering the hexadecimal ID or name of the journal. | | | |
| Input | **Input name** | **DateType** | | **Valid values conditions** |
| identification | String | | Maximum 100 characters with spaces are accepted |
| Result and postcondition | The system will review the user’s data to verify if they have this magazine, with the purpose of deleting the active subscription and disabling it, at the end it will update the database and present a message if the purpose of the function was successfully saved or executed or not. . | | | |
| Output | **Output name** | | **DateType** | **Format** |
| msg | | String | A text string message will be displayed, informing if the function was carried out correctly or not. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Indetification and name | FR7. Show product bibliographic of the user | | | |
| Summary | The system will display the user's bookshelf with a size of 5\*5, allowing them to navigate to the next or previous shelf if possible. The user can also select a book to read by using its code or coordinates within the bibliographic collection | | | |
| Input | **Input name** | **DateType** | | **Valid values conditions** |
|  | **wordKey** | **String** | | **Accept:**  S. Next Page  A. Back Page  B. Back Menu User  Code  Coordinates: x-y |
| Result and postcondition | If the option is valid, the system will recognize the user's selection. If the user chooses "A" or "S," it will display other bookshelves. If "B" is chosen, it will show the main menu. Lastly, if it is not a valid input yet, the system will wait for a correct input value. | | | |
| Output | **Output name** | | **DateType** | **Format** |
| showShelves(A o S) | | String | Displays a 5\*5 matrix of the user's library shelves |
| ShowRequirements 8 (coordinates or code) | | String | show the following method |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Indetification and name | **FR8. Simulate reading of the bibliographic product,** | | | |
| Summary | The software will allow simulating the reading session of a bibliographic product by console, showing the name of the product, the current page and navigation options to read the previous page(A), the next page(S) and to return to the Library( D). Also, each page read in the simulation will increase the number of pages read for the product on the platform. For regular users, banner ads will be displayed at the beginning of the reading session and after every 20 pages read in a book or 5 pages read in a magazine. | | | |
| Input | **Input name** | **DateType** | | **Valid values conditions** |
| optionRead | Char | | Only these options are accepted:  A. Previous page  S. Next page  B. Return to the library |
| Result and postcondition | The software will change the position of the page according to the option chosen by the user, if it returns to the library it will return the pages read to accumulate them in the product (it will have a verifier if it has really read pages and that it has not only been returned to read them ) | | | |
| Output | **Output name** | | **DateType** | **Format** |
| msg | | String | If you quit reading, it returns a message “How did you like the book? And what do you plan to do now? As a rhetorical question |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Indetification and name | **FR8. Report Generation** | | | |
| Summary | This control method focuses on generating reports that the user wants, such as:  1. Provide the total accumulated number of pages read across the platform for each type of bibliographic product, including books and magazines (product type and number of pages read).  2. Report the most popular genre for books and category for magazines across the platform, including the genre/category name and the number of pages read.  3. Display the top 5 most read books and the top 5 most read magazines on the platform, including the book/magazine name, genre/category name, and number of pages read.  4. For each genre, inform the number of books sold and the total sales value ($).  5. For each category, provide the number of active subscriptions and the total amount paid for subscriptions. | | | |
| Input | **Input name** | **DateType** | | **Valid values conditions** |
| option | Int | | Only these options are accepted:  [1-5] |
| Result and postcondition | The System will identify the option chosen by the user, for which it will show and execute the corresponding function, showing the desired message in the application | | | |
| Output | **Output name** | | **DateType** | **Format** |
| msg | | String | A text string with the function chosen by the user of the system |

# TRACEABILITY TABLE

|  |  |  |  |
| --- | --- | --- | --- |
| **Functional Requirements** | **Class name** | **Method name** | **Description** |
| **FR0. Register Users** | Main | registerUser | This function registers a user by prompting for their name, unique ID, and user type (regular or premium) and then calling the registerUser method in the controller class. |
| Controller | registerUser | This control method is for register the users, according if they are premium or regular |
| **FR1. Register Bibliographic Products** | Main | registerBibliographicProduct | This method records the bibliographic products, according to the type of book that the user wants to appear options for such own regitters |
| Controller | registerBibliographicProduct | This control method create and add a new biblipgraphic product |
| generationAlfaAndHexaDecimal | This control method is for generate a unique code Hexadecimal or alphanumeric to bibliographic product |
| assignTypeBook | This method is for assign the type that is theBook |
| assignTypeEmission | This method is for assign the type emission of the Magazine |
| assignTypeMagazine | This method is for assign the type that is the Magazine |
| **FR2. Modify Bibliographic Product** | Main | modifiedBibliographicProduct | This view method is for Modify bibliographic products |
| Controller | modifiedProductBibliographic | This control method create and add a new biblipgraphic product |
| searchBibliographic | This control method is for search the bibliographicProduct by its name or id |
| assignTypeBook | This method is for assign the type that is theBook |
| assignTypeEmission | This method is for assign the type emission of the Magazine |
| assignTypeMagazine | This method is for assign the type that is the Magazine |
| Bibliographic | Setters | Save new dates |
| Book | Setters | Save new dates |
| Magazine | Setters | Save new dates |
| **FR3. Delete Bibliographic Product** | Main | deleteBibliographicProduct | This view method is for delete bibliographic products |
| Controller | searchBibliographic | This control method is for search the bibliographicProduct by its name or id |
| deleteProduct | This control method eliminates the product in the main library, but not the users who already bought it |
| Bill | setProduct | This method makes a backup when the disagreement system a product, so the product that the user bought was not deleted from the system |
| positionProduct | This Java function returns the position of a bibliographic product in a list based on its name |
| **FR4. Program Starter Text** | Controller | testInit | The function initializes and populates various objects such as users, bibliographic items, and bills, and returns a formatted string displaying the information. |
| **FR5. Allow user to purchase bibliographic products** | Main | BuyBibliographicProduct | This method of view focuses on receiving and entering the information to proceed to the purchase of bibliographic products from a specific user |
| Controller | searchBibliographic | This control method is for search the bibliographicProduct by its name or id |
| intanceOfBibliographic | This control method is for decide the type intance the of bibliographic product |
| CheckingCheck | This method verifies if the user can buy 1.Boor or 2.Magazine if it is a regular user, if it is Premium, it passes the test, it means that it can. |
| Regular | counterProduct | This method is responsible for counting the products bought from the regular user, with the purpose of not getting more of their parameters |
| User | alreadyHasProduct | This method verifies if there is already the bibliographic product in the user library |
| Bibliographic | getData (Abstract ) | This method has the function of obtaining the information of the bibliographic product. |
| getValue | Return the value of the product |
| **FR6.Allow User to Cancel Subscription** | Main | eliminateMagazineSubscription | The view method is to eliminate subscriptions with user magazines |
| User | alreadyHasProduct | This method verifies if there is already the bibliographic product in the user library |
| eliminateMagazineSuscription | This user's method is responsible for eliminating the subscription of a magazine that has in your library |
| Controller | intanceOfBibliographic | This control method is for decide the type intance the of bibliographic product |
| eliminateMagazineSubscrition | This method is responsible for eliminating the subscription of a magazine |
| Bill | eleminateMagazine | his method eliminates the subscription of a magazine |
| positionProduct | This Java function returns the position of a bibliographic product in a list based on its name |
| **FR7** **Show product bibliographic of the user (Temporary until library presentation is enabled)** | User | showProduct | This function generates a formatted string displaying the products (magazines and books) in the bills list. |
| organizeBibliographicProducts | This function organizes the user's bibliographic products by collecting them from bills, sorting them by date, and linking them in a matrix. |
| productSearched | This function searches for a bibliographic product in a library based on a given code or coordinates |
| alreadyHasProduct | This method verifies if there is already the bibliographic product in the user library |
| Bill | getproduts | This method returns the bibliographic product that is related to the invoice |
| Bibliographic | getCodeId | Return the products name |
| **FR8. Simulate Reading of Bibliographic Product** | Main | read | This method is responsible for simulating the sensation of reading  any bibliographic product that the user possesses |
| User | alreadyHasProduct | This method verifies if there is already the bibliographic product in the user library |
| getBill | This method focuses on searching and returning the Bill object where the bibliographic product has the user |
| Controller | read | The function reads a section of a product and returns a message with options for the user. |
| Bill | increasePages | The function increases or decreases the page number of a given bibliographic product based on the user's input and returns the updated page number. |
| positionProduct | This Java function returns the position of a bibliographic product in a list based on its name |

# Design UML (Down)

