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

| CLIENT | ReadX |
| --- | --- |
| USER | General poblation |
| FUNCTIONAL REQUIREMENTS | R1. Register premium and regular users.  R2. Register new products  R3. Modify books and magazines.  R4. Delete books and magazines.  R5. Generate objects for each type of user and product.  R6. Buy a book.  R7. Subscribe to a magazine.  R8. Simulate a lecture session. |
| CONTEXT | The Egyptian company ReadX requires software that allows them to manage their business worldwide.  The software offers bibliographic products, initially only magazines and books. There are 3 kinds of magazines: Variety, design, and scientific. Also, there are 3 genres of books: Science fiction, fantasy, and historic novel.  The system must allow any kind of user to buy books or be subbed to any magazine.  The user chooses to stay in the regular version or upgrade to the premium one. Regular users are forced to watch ads and are not able to buy more than 5 books and be subscribed to 2 magazines ( The prices are in Dollars). The system must save the name, ID, and linking date, of every user.  The system must have a library that shows the products owned by the user, this has to be done using a 5x5 matrix. Also, it should allow the user to do a lecture simulation that contains navigation options throughout the process. |
| NON-FUNCTIONAL REQUIREMENTS | R1. The interface must be intuitive.  R2. It should contemplate the creation of future types of users.  R3. It should contemplate the creation of future kinds of products.  R4. Should use a graphic interface.  R5. The prices must be displayed in dollars. |

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

| Name | R1. Register premium and regular users. | | |
| --- | --- | --- | --- |
| Resume | The system must give the user the option to be either a regular or premium user. Regular users are allowed to buy 5 books and sub to 2 magazines. It also needs to ask for the name, ID, and linking date, in order to register the user. | | |
| Entries | Entry name | Data type | Selection Condition |
| name | String | N/A |
| id | String | Different from the others |
| linkingDate | Calendar | Actual time |
| Result | The user data is saved in the database. | | |
| Exits | Entry name | Data Type | Selection Condition |
| confirmationMsg | String | The register should be successful. |

| Name | R2. Register new Product | | |
| --- | --- | --- | --- |
| Resume | The system will receive all of the following data to register the product: ID, name, number of pages, publication date, gender, category, quick review, URL that takes the user to a repository with the portrait, price, number of copies sold, periodicity of emission, active subscriptions, and amount of read pages. It also will receive a number that decides what kind of product is going to be created whether it is a book or a magazine. | | |
| Entries | Entry name | Data type | Selection Condition |
| flag | int | Decides which object to create. |
| IdMagazine | String | 3 hexadecimal characters |
| idProduct | String | 3 alphanumeric characters |
| name | String | N/A |
| gender | Gender | Science fiction, fantasy or historic novel. |
| category | Category | Variety, design, or science. |
| URL | String | Should lead to the portrait |
| price | double | In dollars. |
| readPages | int | N/A |
| activeSubscriptions | int | N/A |
| emisionPeriodicity | String | Monthly, daily, or annually. |
| copiesSold | int | N/A |
| shortReview | String | N/A |
| numberOfPages | int | Different from 0 |
| publicationDate | Calendar | Real-time |
| Result | The product data is saved in the system | | |
| Exits | Entry name | Data Type | Selection Condition |
| confirmationMsg | String | The product should be registered. |

| Name | R3. Update products. | | |
| --- | --- | --- | --- |
| Resume | The system should receive the name of the product that is going to be modified, and then ask for the new data that is going to be saved. Not all of the data can be changed, only the following have the option to be changed: Name, gender, category, URL, price, active subscription, periodicity of the emission, number of copies sold, the short review, and the number of pages. | | |
| Entries | Entry name | Data type | Selection Condition |
| name | String | registered Previously |
| gender | Gender | Science fiction, fantasy, or historic novel. |
| category | Category | Variety, design, or science. |
| URL | String | Should lead to the portrait |
| price | double | In dollars. |
| activeSubscriptions | int | N/A |
| emisionPeriodicity | String | Monthly, daily, or annually. |
| copiesSold | int | N/A |
| shortReview | String | N/A |
| numberOfPages | int | Different from 0 |
| Result | The new data is saved in the system. | | |
| Exits | Entry name | Data Type | Selection Condition |
| confirmationMsg | String | N/A |

| Name | R4. Delete Products | | |
| --- | --- | --- | --- |
| Resume | The system should receive the name of the product, and then it should look for it in the array that contains it, and replace it with a null. | | |
| Entries | Entry name | Data type | Selection Condition |
| nameProduct | String | Registered previously |
| Result | The object is replaced with a null. | | |
| Exits | Entry name | Data Type | Selection Condition |
| confirmationMsg | String | N/A |

| Name | R5. Generate objects for each type of user and product. | | |
| --- | --- | --- | --- |
| Resume | In order to test the functionalities, the system should give the option to create an object of each type with previously defined attributes. | | |
| Entries | Entry name | Data type | Selection Condition |
| N/A | N/A | N/A |
| Result | The objects are created. | | |
| Exits | Entry name | Data Type | Selection Condition |
| confirmationMsg | String | The objects had to be created. |

| Name | R6. Buy a book | | |
| --- | --- | --- | --- |
| Resume | The system should receive the name of the user and the name of the book. Then the book is added to the user's inventory that is defined as an array. Also, it should store the date when this operation was made, the price that was paid, and should update the counter of copies sold. | | |
| Entries | Entry name | Data type | Selection Condition |
| userName | String | Must be registered previously |
| bookName | String | Must be registered previously. |
| Result | The book is added to the inventory of the user | | |
| Exits | Entry name | Data Type | Selection Condition |
| confirmationMsg | String | N/A |
| pricePaid | String | The price of the book |

| Name | R7. Subscribe to a magazine. | | |
| --- | --- | --- | --- |
| Resume | The system should receive the name of the user and the name of the magazine. Then the magazine is added to the user's inventory which is defined as an array. Also, it should store the date when this operation was made, and the price that was paid, and should update the counter of active subscription. | | |
| Entries | Entry name | Data type | Selection Condition |
| userName | String | Must be registered previously |
| magazineName | String | Must be registered previously. |
| Result | The magazine is added to the inventory of the user | | |
| Exits | Entry name | Data Type | Selection Condition |
| confirmationMsg | String | N/A |
| pricePaid | String | The price of the book |

| Name | R8. Simulate a lecture session. | | |
| --- | --- | --- | --- |
| Resume | The system should receive the name of the user and the name of the book. Then it verifies that the user owns that book, if it does, then it proceeds to start the lecture session that displays the current page that is being read and also gives the option to return to the previous page or to go forward. Until it chooses to go back to the original menu. | | |
| Entries | Entry name | Data type | Selection Condition |
| userName | String | Must be registered previously |
| bookName | String | Must be registered once. |
| Result | The simulation begin | | |
| Exits | Entry name | Data Type | Selection Condition |
| confirmationMsg | String | N/A |
| pricePaid | String | The price of the book |

| Name | R8. Simulate a lecture lesson. | | |
| --- | --- | --- | --- |
| Resume | The system receives the name of the user and the book or magazine to read, shows the name of the bibliographic product, the current page being read, and navigation options to read the previous page, the next page and to return to the Library. It also should increase the number of pages read for the corresponding bibliographic product on the platform. And if the user is regular ads should be presented at two times: at the beginning of the reading session and after every 20 pages read in a book or 5 pages read in a magazine. | | |
| Entries | Entry name | Data type | Selection Condition |
| productName | String | Should be in the user inventory |
| userName | String | N/A |
| Result | The simulation is shown. | | |
| Exits | Entry name | Data Type | Selection Condition |
| currentPage | String | Real page |
| nextPage | String | Option 1 |
| previousPage | String | Option 2 |
| returnLibrary | String | Option 3 |
| productName | String | Should be in the inventory |