In general, the document is well structured, but there are some deficiencies which are addressed below:

**SRS:**

Requirements need more than a simple list of requirements. The great thing is that you have a clear and unambiguous statement for each requirement, but it is not sufficient.

I provide you with a sample possible solution:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Req. ID** | **Requirement** | **Priority** | **Rel. Req.** | **Req. Type** |
| G1 | There are two types of users: Admin and Member. | M | G2, G3, G4 | F | U | PD |
| Description | These two roles (Admin and Member) have different functionalities in the PLS.  Users must login to the system for any operation. | | |  |
| G2 | Users should be able to login into their accounts via the interface. | M | G1, G3, G4 | F | U | PD |
| Description | User interface should not offer any ability to a user, if they are not logged in.  Accounts separate admin and members capabilities. | | |  |
| G3 | No option is available to users, if they are not logged in, yet. | M | G1, G2, G4 | NF | U | PD |
| Description | There could be only ‘login’ and ‘exit’ options available to the users, when the system is just started. | | |  |
| G4 | Admin login credentials are predefined in the system. | M | G2, G3, G4 | NF | U | PD |
| Description | Admin’s username and password are hardcoded in the system, as below:  Username = “admin”  Password = “admin123” | | |  |
| … | … | … | … |  |
| Description |  | | |  |

|  |
| --- |
| F = Functional  U = Users  PD = Product  … |

The sample can give you an idea what could be added to the requirements list. You need to tag them, categorize them, prioritize them, and show the relationship among different requirements. I can see the priority in your list, but it is lacking the other useful information.

I did not fully check for verification and validation of your list of requirements. Make sure that all requirements are in your list. For example, I did not see the limitation that there is no function for a user who is not logged in yet.

**Class diagram:**

The text in the class diagram is not clearly visible. It is blurry.

The required classes are depicted in the diagram, but there are some missing components and inconsistencies. Just few examples:

1. The loan item is composed of member. Firstly, there is no member inside the class loanitem, so how loanitem is composed of member. Secondly, this composition means if there is no loanitem, there won’t be member too. Is this correct?
2. I cannot say much about your personal thoughts about the design, because it depends how you really want to develop the system. but I believe there are some missing relationships. For example, when a library admin lend a book item to a member, you should pass the information of the book item to the method loanbookitem() in libraryadmin (unless I misunderstood your design). If this is correct, then libraryadmin should be related to bookitem.
3. Some multiplicities are missed in the diagram. For example, a member can have a maximum of 3 loaned books.

**Use case diagram:**

There are some inconsistencies in the diagram. For example:

1. Search book should be an independent use case to be called by member, so should be directly connected.
2. View catalog is an extension of search book. It means it will be called by search book, occasionally. But the search book itself is not executed by any actor. The same issues are visible in admin use cases. I guess you are not clear about the direction of the extend. Maybe you can review the slides or references, again.
3. On the other hand, there must be some include and extend which are not in the diagram. For example, if a member wants to loan a book, an availability check use case should be included, and extend use case could be to display a proper message that book item is not available. Or the book item might be still under loan by the same member, so it could not be loaned twice, before returning it.

Please note that not every single part of the work is checked and addressed here in my comments. These are only some of my observations, with some examples and advice to help you improving your work. Furthermore, some parts will become clearer after development. So, I have no idea of the purpose of some of your design components, attributes, or methods.

Please ensure that you check all the requirements as described in the assignment and produce a solid design and code.