**Bahir Dar University**

**Library Management System**

**System Requirement Document**

**Comments**

System & User requirements

1. The way you write user requirements is good but most of the parameters (source, related requirement,….) for each requirements are unfilled
2. **Needs proofreading** - Some grammatical and spelling errors should be avoided
3. Some requirements are non-functional requirements but stated as functional. For example *modifiability , Req. ID 208, Req. ID 209*
4. **Where are the system requirements?**

Class diagram

1. Class diagram is not detailed enough – lacks type of association, multiplicity,
2. Cohesion of the book class is under question
* Is your system for only books? ???
* ….
1. Why librarian can’t be subclass of users?
2. Generally, your class diagram need improvement

Use case diagrams & Descriptions

1. What do you mean by maintain XXXX??? How can a librarian maintain book info (this is job of data base or ….)
2. What is Accusion??
3. Why staffs can’t reserve a book?
4. Who is director? I don’t think each school have its own library.
5. Re-scrutinize relevance of each use case and if there is any use case left
6. Use case descriptions are incomplete, disorganized and carelessly done
7. better to give basic course of actions in two columns with *user action* and *system response* column headings
8. ………

Sequence Diagrams

1. Who supposed to perform the actual returning process?
2. Are these the only sequence diagrams for the system

Non functional requirements

1. Requirements do not have desired/wanted characteristics: testability, clarity, ….
2. How is a User interface non functional requirement?

**Table of Content**

[Functional Requirement 3](#_Toc292528317)

[List of requirements 3](#_Toc292528318)

[Class Diagram 10](#_Toc292528319)

[Use case diagram 11](#_Toc292528320)

[Business Model 12](#_Toc292528321)

[List of use cases 16](#_Toc292528322)

[Detail Use Case Scenario and Sequence diagram 17](#_Toc292528323)

[Non – Functional Requirement 28](#_Toc292528324)

[Workability 28](#_Toc292528325)

[Reliability 28](#_Toc292528326)

[Usability 28](#_Toc292528327)

[Efficiency 28](#_Toc292528328)

[Maintainability 28](#_Toc292528329)

[Portability 29](#_Toc292528330)

[Concurrency 29](#_Toc292528331)

[Error Handling 29](#_Toc292528332)

[User Interface 29](#_Toc292528333)

[Documentation 29](#_Toc292528334)

[Backup 29](#_Toc292528335)

[Security 29](#_Toc292528336)

# Functional Requirement

## List of requirements

 **1**. **User management**

|  |  |  |
| --- | --- | --- |
| Req. ID: 200 | Source: | Priority:  |
| Name: | *User registration*  |
| Description: | The system registers users by taking different information from the user.  |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 201 | Source: | Priority:  |
| Name: | *Librarian registration*  |
| Description: | The system registers a librarian by taking different information from the librarian as if in the case of a normal user plus with additional functionality. |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 202 | Source: | Priority:  |
| Name: | *Delete user’s record*  |
| Description: | The system deletes users when the admin wants the user to be deleted from the database records. |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 203 | Source: | Priority:  |
| Name: | *Delete librarian’s record*  |
| Description: | The system deletes librarian when the admin wants the librarian to be deleted then records available for the given librarian will be deleted from the database of the system. |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 204 | Source: | Priority:  |
| Name: | *Add new user*  |
| Description: | The system allows the admin to add new user record to the database. |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 205 | Source: | Priority:  |
| Name: | *Add new librarian* |
| Description: | The system allows the admin to add new librarian record to the database. |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 206 | Source: | Priority:  |
| Name: | *View librarian info*  |
| Description: | The system allows the admin to view a given librarian info. |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 207 | Source: | Priority:  |
| Name: | *View user info*  |
| Description: | The system allows the admin to view a given user info. |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 208 | Source: | Priority:  |
| Name: | *Modify user’s info*  |
| Description: | The system allows the admin to modify a given user info. |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 209 | Source: | Priority:  |
| Name: | *Modify librarian’s info* |
| Description: | The system allows the admin to modify a given librarian info. |
| Related Req. |  |

 **Search**

|  |  |  |
| --- | --- | --- |
| Req. ID: 100 | Source: | Priority:  |
| Name: | *Search User* |
| Description: | The librarian or the manager of other sections search for Student, Instructor, Staff Member. |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 101 | Source: | Priority:  |
| Name: | *Search book by title* |
| Description: | All user of the system search for specific type of book by using by using its title. |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 102 | Source: | Priority:  |
| Name: | *Search User by Author* |
| Description: | All user of the system search for specific type of book by using by using its Author.  |
| Related Req. |  |

**Weekly reports**

|  |  |  |
| --- | --- | --- |
| Req. ID:  | Source: | Priority:  |
| Name: | *Daily borrowed books*  |
| Description: | The system reports the list of daily borrowed books in their Call No and the User types. |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID:  | Source: | Priority:  |
| Name: | *Daily returned books*  |
| Description: | The system reports the list of daily returned books in their Call No and the User types. |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 103 | Source: | Priority:  |
| Name: | *Search User by Subject* |
| Description: | All user of the system search for specific type of book by using by using its Subject.  |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 104 | Source: | Priority:  |
| Name: | *Search for unreturned book* |
| Description: | The circulation manager searches for the book that is not returned on time. |
| Related Req. |  |

**Borrow book**

|  |  |  |
| --- | --- | --- |
| Req. ID: 400 | Source: | Priority: medium  |
| Name: | *Check book availability*  |
| Description: | The system checks whether the book available in the library or not before the book is borrowed when a user searches the book.  |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 401 | Source: | Priority: medium |
| Name: | *Check books for borrowing*  |
| Description: | The system checks whether the book can be borrowed or not. The system searches the book from the stack. If it is not there then it is not allowed to borrow. Because it may be for reservation or for reference.  |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 402 | Source: | Priority:  |
| Name: | *Check legibility of users*  |
| Description: | The system checks whether the user is legible to borrow the book or not. The user must be under the list of the student, staffs, instructors or other allowed persons so that the system will allow the user to borrow the book.  |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 403 | Source: | Priority:  |
| Name: | *Check user’s maximum allowed books*  |
| Description: | The system checks the number of books that the user is allowed to borrow. If the user reaches its maximum allowed number the system will deny the service.  |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 404 | Source: | Priority:  |
| Name: | *Add book to borrowed list*  |
| Description: | The system adds the lend book to the list of borrowed book for future reference.  |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 404 | Source: | Priority:  |
| Name: | *Add borrower to borrower list*  |
| Description: | The system adds the librarian who borrows that specific book for the user. |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 405 | Source: | Priority:  |
| Name: | *Add user information to issuer list*  |
| Description: | The system adds the user to the list of issuer with the book that he/she borrowed. |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 405 | Source: | Priority:  |
| Name: | *Register borrowed date*  |
| Description: | The system registers the borrowed date of the book for the particular user. Then the return date will be calculated from this date.  |
| Related Req. | 302 |

 **Return Book**

|  |  |  |
| --- | --- | --- |
| Req. ID: 301 | Source: | Priority:  |
| Name: | *Check borrowed book from borrowed list*  |
| Description: | The system should check whether the book is borrowed or not from the list of borrowed books that are registered during borrowing books when a librarian enters the title, author, or ISBN of the book on search box of borrowed books.  |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 302 | Source: | Priority:  |
| Name: | *Check return date*  |
| Description: | The system checks the return date of the book by taking the borrowed date from the borrowed list and calculates the allowed time for the user who borrowed the book.  |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 303 | Source: | Priority:  |
| Name: | *Calculate penalty*  |
| Description: | If the user passes the deadline for returning the book, the system calculates the payment for the days after the return date. In each day there is penalty of 0.25 cents per day which result n\*0.25 birr for n days. |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 304 | Source: | Priority:  |
| Name: | *Update user status*  |
| Description: | The system updated the status the user who returned the book on time.  |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 305 | Source: | Priority:  |
| Name: | *Update book status*  |
| Description: | The system updated the status the book if it returned to the stack so that it could be available for next lending.  |
| Related Req. |  |

|  |  |  |
| --- | --- | --- |
| Req. ID: 306 | Source: | Priority:  |
| Name: | *Calculate cost*  |
| Description: | The system calculates the total cost the book that should be paid by the user if he/she lost the book. |
| Related Req. |  |

## Class Diagram



## Use case diagram



Staffs can’t reserve book? Why?

## Business Model

In this section we have try to show the activities in each department and sub department of the library performed. We are trying to show it using the business use cases and the corresponding activity diagram.

**Table Business use case Models**

|  |  |  |
| --- | --- | --- |
| Use case id  | name  |  description |
| BUC -001 | Search  | In this use case every user of the library system are searching for books, periodicals, users. |
| BUC -002 | Manage Periodicals | In this use case new periodicals are registered, changed periodical information will updated. |

|  |  |
| --- | --- |
| **Use case Id:** | **BUC-001** |
| Use Case Name: | Search |
| Description | In this use case users of the library system will searching books, periodicals and users. |
| Actors: | Primary Actors:* Users.
* Librarians.
 |
| Preconditions: | Users and librarians should be authorized. |
| Triggers: | * Whenever the user or the librarian want to search.
 |
| Flow of Events: | 1. Users or librarians login to the system.
2. Users or librarians search what they want.
3. Users or librarians get what they want.
 |
| Post condition: |  The user or the librarian gets the information what they want. |
| Alternate Flow: | None |
| Exceptions: | There is no exception to search other that the authorized user. |
| Information Requirements: | * Book title
* Book callNo, AccNo,ISBN.
* Book author,Publisher.
* Book status( borrowed, available)
* Periodicals
* Borrowed users
* Users.
* Shelf No
* Librarians.
 |
| Assumptions: | All information seekers are authorized. |
| Notes and Issues: | When we are saying users it includes students, lectures, researchers, staff members. |
| Activity Diagram: |  |

|  |  |
| --- | --- |
| **Use case Id:** | **BUC-002** |
| Use Case Name: | Manage Periodicals |
| Description | In this use case the librarians add new periodicals and update, delete, and add new periodicals information.  |
| Actors: | Primary Actors:* Librarians.
 |
| Preconditions: |  Librarian should be authorized. |
| Triggers: | * Whenever there is new information about the a periodical.
* Whenever there is new periodical
 |
| Flow of Events: | 1 Librarians login to the system.1. Librarians register new periodical.

3 Librarians get periodical registered. |
| Post condition: |  The user or the librarian gets the information what they want. |
| Alternate Flow: | * 1. librarian want to update periodical information
	2. librarian search periodical
	3. librarian update periodical information
	4. librarian get updated information
	5. librarian want to delete periodical
1. librarian search periodical
2. librarian delete periodical
3. librarian get periodical deleted
 |
| Exceptions: | There is no exception to search other that the authorized user. |
| Information Requirements: | * periodical title
* periodical type
* periodical number
* Periodical author, Publisher.
 |
| Assumptions: | The librarian is authorized. |
| Notes and Issues: | In periodicals it includes fictions.  |
| Activity Diagram: |  |

## List of use cases

List of System Use Cases (SUC)

| **No** | **SUC ID** | **SUC Name** | **SUC Description** |
| --- | --- | --- | --- |
| 1 | SUC-001 | Borrow books  | This use case describes the overall process of borrowing books for students, instructors, staffs and some legible external users. Librarian on the circulation section are takes the complete process of this use case. For each circulation works there is independent account for handling authorization cases. |
| 2 | SUC-002 | Return book  | This system process explains the whole activities involved during returning the book to the stack. After the user has borrowed the book it is expected to return the book on time. The system checks the deadline for returning the book and if there is penalty it calculates the fee. |

|  |  |  |
| --- | --- | --- |
| SUC -001 | Search  | In this use case the system will search books, periodicals, users. |
| SUC -002 | Manage Periodicals | In this use case the system will give manage periodicals service. |

| **No** | **SUC ID** | **SUC Name** | **SUC Description** |
| --- | --- | --- | --- |
| 1 | SUC-001 | User management | This use case describes the entire process of handling or managing all users of the system including users and librarian. This may include registering, updating modifying, adding or deleting, to view records of a given user or librarian info. |

## Detail Use Case Scenario and Sequence diagram

1. Search

|  |  |
| --- | --- |
| Use Case ID: | SUC-001 |
| Use Case Name: | search |
| Created By: |  |
| Date Created: | 6-05-2010 |
| Updated By |  |
| Date of Update |  |
| Actors: | Primary Actors* Librarian
* users
 |
| Description: | This use case searches books, periodicals, users |
| Trigger: | The user or librarian searches for users, books, periodicals |
| Preconditions: | User or librarians should be authenticated. |
| Basic Flow: | 1. The user or librarian enter the string he want to search
2. The user selects searching category
3. The user press enter the searching button
4. The system will display the search result
 |
| Post conditions: | Search result will displayed  |
| Alternative Flows: |  |
| Exceptions: |  |
| Priority: | Intermediate |
| Frequency of Use: |  |
| Business Rules: |  |
| Special Requirements: | The page should be user friendly.The searching method should be user friendly and ease to use.The system have to take maximum of 2 sec. to search. |
| Assumptions: |  |
| Notes and Issues: |  |

Manage periodical

|  |  |
| --- | --- |
| Use Case ID: | SUC-002 |
| Use Case Name: | Manage Periodicals |
| Created By: |  |
| Date Created: | 6-05-2010 |
| Updated By |  |
| Date of Update |  |
| Actors: | Primary Actors* Librarian
 |
| Description: | This use case manages periodicals |
| Trigger: | The librarian selects manage periodicals |
| Preconditions: | User or librarians should be authenticated. |
| Basic Flow: | 1. The librarian selects the “manage “button
2. The system will display the manage periodicals window
3. The user selects registration menu
4. The user fills the form
5. The use submit the form
6. The system checks the inputs
7. The system will display successful message
 |
| Post conditions: | The periodical will registered  |
| Alternative Flows: | 3.1 the user selects update periodicals menu 1. the user select the update periodicals menu 2. the system will display the update periodical window 3. the user update the information what he want  4. the user submit the updates 5 the system checks the inputs  6 the system will display success message**Post condition :**  the system update the periodicals information3.2 the user selects delete periodicals menu 1. the user selects the delete periodicals menu 2. the system will display the delete periodicals window 3. the user selects the periodical to delete 4. the user delete the periodical what he want 5. the system will display deleted successfully message**Post condition :** the system delete the periodical7.1 the system get invalid input 1. the system will display error message with some description 2. go to Basic course of action 3  |
| Exceptions: |  |
| Priority: | Intermediate |
| Frequency of Use: |  |
| Business Rules: |  |
| Special Requirements: | The page should be user friendly.The searching method should be user friendly and ease to use.The system should take maximum of 2 sec. to search.The system should take maximum of 1 sec to delete.The system should take maximum of 2 sec to update.The system should take maximum of 5 sec to register. |
| Assumptions: |  |
| Notes and Issues: |  |

Manage user

|  |  |
| --- | --- |
| Use Case ID: | SUC-002 |
| Use Case Name: | Manage users |
| Created By: | Nebiyou.S |
| Date Created: | 01/05/2011 |
| Updated By |  |
| Date of Update |  |
| Actors: | * admin
* users(students, instructors, stuffs, librarian)
* issuer
 |
| Description: | This use case describes how users of the system are managed.  |
| Trigger: | The system needs to be legal or being managed by an admin. |
| Preconditions: | Admin is selected for managing the system. |
| Basic Flow: | 1. admin wants to view users or librarian info then clicks view button to see
2. The system displays user or librarian info.
3. Admin wants to modify users or librarian info then clicks update button to change info.
4. System updates info with the new record.
5. The admin wants to add a new user to the system
6. The system provides all necessary procedures
7. Admin fills the form and adds the user to the system.
8. Admin wants to delete a given user / librarian info from the database.
9. Admin selects the given user/ librarian and clicks on delete button.
10. The system deletes the selected user/ librarian and associated info with them.

  |
| Post conditions: | The system handles the new state with what the admin sets. |
| Alternative Flows: | 1. The system shows a conformation message. |
| Exceptions: |  |
| Priority: | High  |
| Frequency of Use: | Minimum of 1/day  |
| Business Rules: |  |
| Special Requirements: | * Security of the system.
* The system should display errors.

The system should be integrated with other automated systems of the Library Management System.  |
| Assumptions: | An authorized admin. |
| Notes and Issues: |  |

Borrow book

|  |  |
| --- | --- |
| Use Case ID: | SUC-001 |
| Use Case Name: | Borrow book |
| Created By: | Mulugeta Ayele |
| Date Created: | 28-08-2003 |
| Updated By |  |
| Date of Update |  |
| Actors: | * Students
* Instructors
* Staffs
 |
| Description: | This use case describes how legible users can borrow books.  |
| Trigger: | The issuer click borrow book link  |
| Preconditions: | The circulation worker must be authenticated.  |
| Basic Flow: | 1. The circulation workers clicks “Borrow book” link
2. The system displays the borrow form
3. The issuer types the user ID
4. The system auto complete the user ID
5. The issuerclicks “Check user legibility” button
6. The system fills users necessary information
7. The issuer fills book information and the system displays the issuer information.
8. The issuer clicks “Borrow” button
9. The system validates the inputs
10. The system displays confirmation message
11. The system sends detail book info to the counter
12. The counter confirms the message when the book is out from the library.
 |
| Post conditions: | The book will be added to the list of borrowed books and the user added to the list of issuers.  |
| Alternative Flows: | * 1. The system shows “No such user” message
	2. The system show “User illegible” message
 |
| Exceptions: |  |
| Priority: | High  |
| Frequency of Use: | Minimum of 2/day  |
| Business Rules: |  |
| Special Requirements: | * Easy for use and secure process
* The system should display errors.

The system should be integrated with other automated systems of the Library Management System.  |
| Assumptions: |  |
| Notes and Issues: |  |



Return Book

|  |  |
| --- | --- |
| Use Case ID: | SUC-001 |
| Use Case Name: | Return book |
| Created By: | Mulugeta Ayele |
| Date Created: | 28-08-2003 |
| Updated By |  |
| Date of Update |  |
| Actors: | * Students
* Instructors
* Staffs
 |
| Description: | This use cases shows the process of returning the book  |
| Trigger: | The circulation worker clicks “Return book” link  |
| Preconditions: | The circulation worker should be authorized  |
| Basic Flow: | 1. The circulation workers clicks “Return book” link
2. The system displays search box to search borrowed books or borrowers or lists some of 10 ..
3. The issuer types the either user ID, book title, call no or author of the book
4. The system displays complete information of borrowed book and borrower.
5. The system checks the return date with returned date
6. The issuerclicks “Return” button
7. The system changes book status to ‘returned’ from borrowed book list.
8. The system displays confirmation message
 |
| Post conditions: | The book will be added to the list of borrowed books and the user added to the list of issuers.  |
| Alternative Flows: | * 1. The system displays “No such borrowed book or borrower” message
	2. The system shows notification for passing of return date
 |
| Exceptions: |  |
| Priority: | High  |
| Frequency of Use: | Minimum of 2/day  |
| Business Rules: |  |
| Special Requirements: | * Easy for use and secure process
* The system should display errors.

The system should be integrated with other automated systems of the Library Management System.  |
| Assumptions: |  |
| Notes and Issues: |  |



# Non – Functional Requirement

## Workability

The system should be Suitable for the variety of users. It should be accurate in performing its functions and secured enough from unauthorized access by other users. Moreover, and it should be complete i.e. it should be fully functional in terms of providing all the functions expected it to perform.

## Reliability

The system should be available all hours of a day and 7 days aweek.

The system should be reliable and matured enough in giving its service. It should have a fault tolerance mechanism in which it can recover faster from problems that may occur. The system should support backup in case the original was accidentally damaged or erased.

## Usability

The system should be understandable by the variety of its users from naïve users to amateurs and IT professionals. The interface should be easy to use and be easy to learn, operate and it should provide alternatives to different tasks.

## Efficiency

The system should be efficient and the response time should be minimal. It should be capable of running on minimum hardware requirements and with the almost all dominant operating systems.

The searching should not take more than 3 seconds.

## Maintainability

The System should be easily maintainable in case of problems, changeable if the need of incorporating new components arises from either the client’s side or technological changes.

## Portability

The system should run on windows and Linux operating system. The system should also run on major browsers Internet Explorer, Mozilla Firefox, Opera, Chrome.

The system should be portable in running on different platforms, adaptable with other systems, installable on different machines architectures, and replaceable if the need arises.

## Concurrency

The System should support at least 100 users at a time.

The system should support multiple accesses of users. It should give service to multiple users concurrently.

## Error Handling

The system should handle exceptions and extreme conditions and behave accordingly. It should notify the users about the type and location of exception and appropriate action to be taken.

## User Interface

The interface of the system should be user friendly, that is, it should be understandable, usable and corrective.

## Documentation

The RAD, SDD and Code documents should be delivered to the Client as of the schedule. Moreover, documents, like user manual and help guide lines which support system users to easily understand and use the system should be delivered.

## Backup

The system shall provide a backup and restore mechanism for solving accidental failures.

## Security

The system should be secured from external intruders and internal misuse. It should have a user’s database and should authenticate each user on log in and should grant user to specific services. The confidentiality of the information gathered during requirement analysis and later phases are kept private and used only for the development of the system internally.