**Albukhary International University**

**School of Computing and Informatics**

**Semester 2 Requirement Engineering CCE2233**

**Assignment**

**Title of Assignment:** **Requirements Engineering of Library Management System (LMS)**

|  |  |
| --- | --- |
| Name | Ali Sena Danishwer |
| Student ID | AIU22102370 |
| Class | C\_C1 |
| Lecturer: | Dr .Ts Mohd Zulkifli Mohd Zaki |
| Submission Date | Week 10 (Friday, 30 May  2025, before/at 11:59 pm) |

Table of Contents

[**Introduction** 3](#_Toc199514040)

[**Requirement Elicitation and Documentation** 7](#_Toc199514041)

[Functional Requirements (FRs) 7](#_Toc199514042)

[Non-Functional Requirements (NFRs) 7](#_Toc199514043)

[**Use Case Modelling** 8](#_Toc199514044)

[**MoSCoW Prioritization for Library Management System** 11](#_Toc199514045)

**Requirements Engineering Report: Library Management System (LMS)**

# **Introduction**

Library Management System is an application which refers to library systems. It is used by librarian to manage the library using a computerized system where they can record various transactions like issue of books, return of books, addition of new books, addition of new students etc. Books and student maintenance modules are also included in this system which would keep track of the students using the library and also a detailed description about the books a library contains. With this computerized system there will be no loss of book record or member record which generally happens when a noncomputerized system is used.

# **Project background**

The development of Library Management System (LMS) is necessary to enhance and make available the operational processes of general libraries, digitizing them. Routine manual processes for circulation, student records and catalog management are common at most educational institutions leading to low efficiency rates of information loss and errors. However, LMS would be to automate those basic functions(borrowing and returning of books, catalog management, fine computing using an automated system. It assists librarians, students and faculty in managing library resources efficiently and reliably without human errors and facilitates better response to information.

# Objective of the report

* Stakeholder engagement in functional/non-functional requirements identification and collection of Library Management System(LMS).
* Apply a structured requirements engineering (re) process: elicit, analyze, specification, validate and manage
* Document requirements clearly in documents, Word excles and Trello for sort, tracking and priority.
* Use case diagrams and descriptions to model the system functionalities as well as stakeholders interactions.
* Requirement Categorizing and Prioritizing-using techniques like MoSCoW for Planning/iteration purposes.
* Ensure proper fit at the user need, institutional and technical level to make a successful implementation of LMS.

## **Requirements Engineering Process Applied**

The requirements engineering in the libraries software included a series of organised steps to collect, investigate, document, validate and maintain system requirements. These were the steps that would help you ensure that the systems being developed would accurately address a user’s needs, and would perform according to their expectations.

### **a. Requirements Elicitation**

The phase was aimed at collecting feedback among stakeholders with respect to what they wanted the system to achieve, what challenges and work issues were encountered right now, what a perfect system should look like (Ali,2017).

**Stakeholder Interviews:**

Interviews LDPs and students were also interviewed to get first hand information on the current workflow, problems experienced and expected features of the new system.

**Document Review:**

Paper-based records and forms in circulation were reviewed to better describe current processes, such as circulation of books, returns, cataloging, and fine collection.

**System Analysis:**

The study catalogued the features, usability problems, and opportunities for improvement in the available library management software.

### **b. Requirements Analysis**

The collected data was analyzed to categorize and prioritize system requirements based on stakeholder input and system feasibility.

* **Functional Grouping:**Requirements are categorized under functional templates like login, book search user interface, inventory control and report generation.
* **Analysis of Dependance and Priority:**Logical relationships of the requirements were decomposed and user needs or implementation difficulties set priority or levels.

### **c. Requirements Specification**

This step involved converting the analyzed requirements into well-structured documentation to guide system design and development.

* **Use of Microsoft Word:**However, Requirements were developed in a structured format using MS Word for declarative specifications and MS Word to portray tabular functional/non-functional requirements with an ID, Priority and Acceptance criteria.
* **Template-Based Documentation:**

Structured templates ensured clear and consistent results. Each requirement was sent an array of the following, namely Title, description type, priority rationale, and result that is expected.

### **d. Requirements Validation**

Validation activities confirmed that the recorded requirements aligned with users' expectations and system constraints.

* **Stakeholder Review Meetings:**Word docs shared with librarians and student representatives the drafted requirements. Feedback was collated and revisions were made thereafter.
* **Checklist-Based Evaluation:**

A validation checklist is used to verify that all tasks, concepts, and requirements are completely covered by true implementation.

### **e. Requirements Management**

Requirements were continuously monitored, updated, and controlled throughout the development cycle to accommodate changes.

* **Use of Trello for Tracking:**I used Trello Spreadsheets to track and manage each requirements status Requirement Title ID Current Status Responsibility (t e m p) Columns — requirement id/proposed or
* **Change Management:**Software (Manually using files with updated version and change logs as Microsoft Word documents) to maintain version control of the modifications and stakeholder approvals.

**3. Requirement Engineering Tool Used**

**Tool Chosen**: **Trello**

* Used for categorizing user stories (e.g., "To Do", "In Progress", "Done")
* Attached requirement documents and notes to cards
* Used labels for priority and stakeholder mapping

# **Stakeholder Analysis**

Stakeholder analysis is a critical component of requirement engineering, as it identifies all parties involved in or affected by the system and outlines their specific needs and expectations. A comprehensive understanding of stakeholders ensures that the final system is both functional and user-centered (Sommerville, 2011).

**List of Identified Stakeholders**

For a Library Management System, the primary stakeholders include:

|  |  |
| --- | --- |
| No | Stakeholders |
| 1 | Library Users (Students, Faculty, and General Public) |
| 2 | Librarians |
| 3 | Library Administrators |
| 4 | IT Support Staff |
| 5 | System Developers |
| 6 | Institutional Management (e.g., University or School Administration) |
| 7 | Vendors or Publishers |
| 8 | Government or Regulatory Bodies |

**Description of each stakeholder’s Needs or Interests**

|  |  |
| --- | --- |
| **Stakeholder** | **Needs/Interests** |
| Library Users | We need a user-friendly interface that makes it easy to search for, borrow, and return books. It should also provide access to digital resources and have a notification system for due dates and new arrivals. Users expect the system to be highly available with minimal downtime. |
| Librarians | Looking for effective tools to help with cataloging, issuing, returning, and managing inventory? It's also important to have reporting and data analytics features to keep track of overdue books and usage statistics. |
| Library Administrators | Interested in system-wide oversight, user management, budget control, and compliance with institutional policies. They require dashboards for performance metrics and resource allocation. |
| IT Support Staff | To keep the system running smoothly, it's essential to have thorough documentation, logs, and configuration tools. This helps us maintain security and tackle any technical issues that may arise. |
| System Developers | To effectively design, implement, and maintain the LMS, we need to gather clear and comprehensive functional and non-functional requirements. It's also essential to get feedback from stakeholders to support our iterative development process. |
| Institutional Management | Interested in alignment with educational goals, cost-effectiveness, student satisfaction, and integration with other institutional systems like student portals or ERP systems. |
| Vendors or Publishers | Require integration capabilities (e.g., APIs) for digital content distribution, licensing, and updates to catalog entries. |
| Government or Regulatory Bodies | Require compliance with data protection laws, accessibility standards, and reporting for educational audits or funding purposes. |

Understanding these stakeholders and their different requirements is important for designing a robust and inclusive library management system. Prioritizing these needs ensures higher user satisfaction, operational efficiency, and compliance with institutional and legal frameworks (Pressman & Maxim, 2014)

# **Requirement Elicitation and Documentation**

Functional Requirements (FRs)  
Based on the attached table, here is all the functional requirements for a library management system.

|  |  |  |  |
| --- | --- | --- | --- |
| Req ID | Requirement Description | Priority | Source |
| FR-01 | The LMS should store all information about librarians and patrons, including their access keys, priority, etc. | High | System Analyst / User Interview |
| FR-02 | The LMS should store all information about items and patrons in two separate databases. | High | Database Design Document |
| FR-03 | The LMS should allow searching items by author, title, or keywords | Medium | User Requirements Document |
| FR-04 | The LMS should allow librarians to add, delete, and modify items in the database, and check the availability of the items | High | Librarian Uses Case |
| FR-05 | The LMS should generate request reports for librarians every day, on the basis of which librarians could make decisions about acquiring or retiring the items. | Medium | Management Requirements |

## Non-Functional Requirements (NFRs)

Based on the attached table, here is all the non-functional requirements for a library management system.

|  |  |  |  |
| --- | --- | --- | --- |
| Req ID | Requirement Description | Priority | Source |
| NFR-01 | Search results should be displayed within 3 seconds | Medium | System Admin |
| NFR-02 | System should support 500 concurrent users | High | Project Brief |
| NFR-03 | Daily backups must be stored securely | High | IT Policy |

## **Constraints**

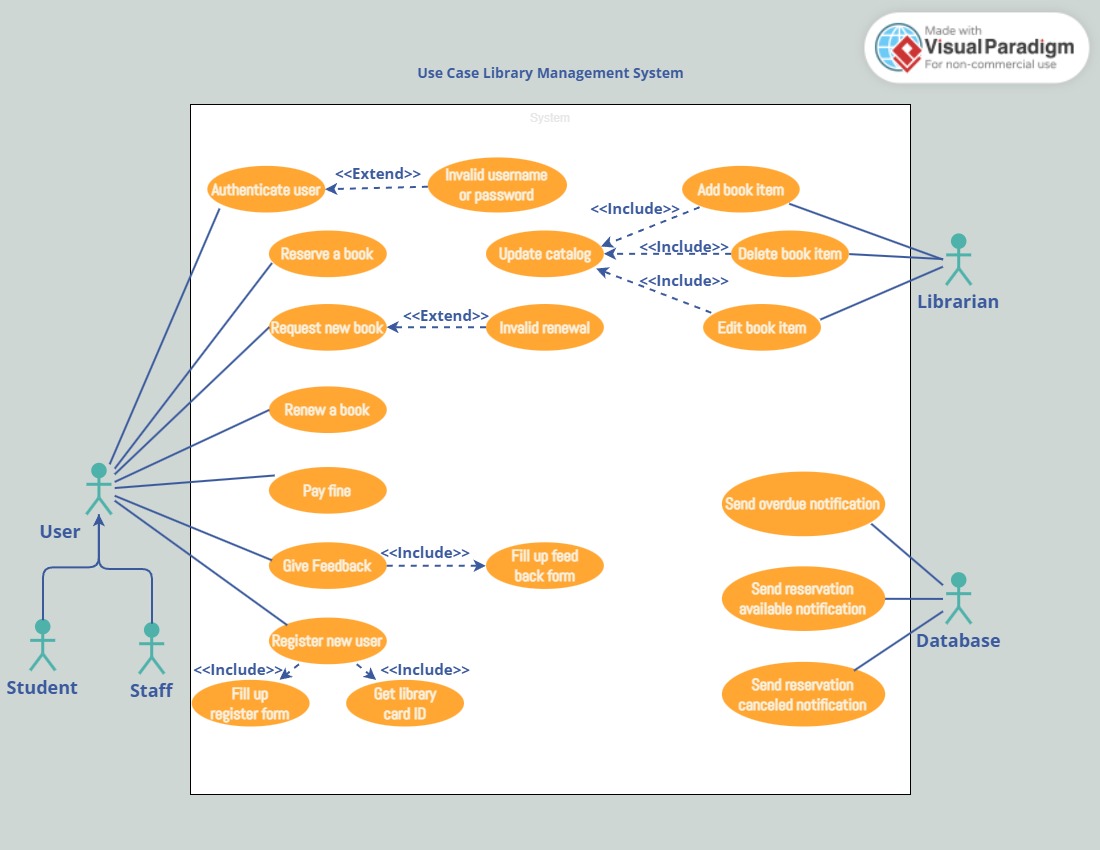
|  |  |  |  |
| --- | --- | --- | --- |
| Req ID | Constraint Description | Priority | Source |
| CR-01 | The system must support up to 500 concurrent users. | High | Non-Functional Requirement – Project Brief |
| CR-02 | Features like RFID tracking and external library integration are out of scope due to budget constraints. | High | MoSCoW Prioritization Table |
| CR-03 | The system must comply with data protection laws and accessibility standards. | High | Stakeholder Analysis – Government or Regulatory Bodies |

# **Use Case Modelling**

The following steps show how to visually map out the relationships and interactions and also the textual description of what the diagram would look like:

|  |  |
| --- | --- |
| 1. Actors | User (Staff or Student)  Librarian |
| 2. Use Cases | Register New User  Issue Library Card  Request New Book  Reserve Book  Renew Book  Pay Fine  Fill Feedback Form  Manage Records  Delete Records  Update Database |
| 3. System Boundary: | The system boundary will encompass all the use cases mentioned above. |

**Below is the use case diagram of a Library Management System:**



**Explanation of Use Case Diagram of a Library Management System**

I have provided the key steps for use case diagram of the library management system as below:

1. User who registers himself as a new user initially is regarded as staff or student for the library system.
   * For the user to get registered as a new user, registration forms are available that is needed to be fulfilled by the user.
   * After registration, a library card is issued to the user by the librarian. On the library card, an ID is assigned to cardholder or user.
2. After getting the library card, a new book is requested by the user as per there requirement.
3. After, requesting, the desired book or the requested book is reserved by the user that means no other user can request for that book.
4. Now, the user can renew a book that means the user can get a new due date for the desired book if the user has renewed them.
5. If the user somehow forgets to return the book before the due date, then the user pays fine. Or if the user forgets to renew the book till the due date, then the book will be overdue and the user pays fine.
6. User can fill the feedback form available if they want to.
7. Librarian has a key role in this system. Librarian adds the records in the library database about each student or user every time issuing the book or returning the book, or paying fine.
8. Librarian also deletes the record of a particular student if the student leaves the college or passed out from the college. If the book no longer exists in the library, then the record of the particular book is also deleted.
9. Updating database is the important role of Librarian (GeeksforGeeks, 2024).

# **Requirement Prioritization for Library Management System**

**MoSCoW technique:**

The table below presents a MoSCoW prioritization plan for a Library Management System (LMS), structured across four iterations. Each use story is categorized by priority, assigned a point value, and includes the source of the requirement. This format also suggests one or more 'Won't have' stories to reflect features out of scope for the current release.(Marthasari, n.d.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iterations | Use Stories | MoSCow Priority | Point | Sources |
| Iteration 1 | Registering the members | Must have | 1 | Librarian request |
| Registering the books | Must have | 1 | Librarian request |
| Inputting the book borrowed | Must have | 1 | System requirement |
| Automated reminder messages | Must have | 1 | User feedback |
| Out-of-scope: Integration with external libraries | Won’t have | 0 | Stakeholder decision |
| Velocity | | | 4 |  |
| Iteration 2 | Finding the books based on the title | Must have | 1 | User feedback |
| Counting the fine automatically | Must have | 1 | Library policy compliance |
| Accepting the notification of the new books | Must have | 1 | Librarian request |
| Observing the due date of borrowing by the admin | Must have | 1 | Admin requirement |
| Out-of-scope: Real-time book tracking via RFID | Won’t have | 0 | Budget constraint |
| Velocity | | | 4 |  |
| Iteration 3 | Registering the new admin | Must have | 1 | System requirement |
| Special access for a user | Must have | 1 | Admin requirement |
| Returning the books | Must have | 1 | Librarian request |
| Changing the website information data | Must have | 1 | UX improvement |
| Velocity | | | 4 |  |
| Iteration 4 | Suggesting the new books | Should have | 1 | User suggestion |
| Uploading the book file | Should have | 1 | Admin request |
| Observing the new books | Could have | 1 | System feature |
| Observing the due date of borrowing | Should have | 1 | Admin suggestion |
| Out-of-scope: Support for multilingual metadata | Won't have | 0 | Project scope limit |
| Velocity | | | 4 |  |

# **Traceability Metrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req ID | Requirement Description | Module / Feature | Test Case / Verification | Status |
| FR-01 | Store all librarian and patron data, including access keys and priority | User Management Module | TC01 | Covered |
| FR-02 | Store item and patron data in separate databases | Data Management Module | TC02 | Covered |
| FR-03 | Allow item search by author, title, or keywords | Search Module | TC03 | In Progress |
| FR-04 | Add, delete, modify, and check item availability | Book Management Module | TC04 | Not Covered |
| FR-05 | Generate daily request reports for acquisition decisions | Reporting Module | TC05 | Covered |
| NFR-01 | Search results must be displayed within 3 seconds | Performance Requirement | TC06 | Pending |
| NFR-02 | Support for 500 concurrent users | Scalability/Technical Design | TC07 | Pending |
| NFR-03 | Daily backups must be stored securely | Security/Backup Module | TC08 | Pending |
| CR-01 | System must support up to 500 concurrent users | Infrastructure Requirement | TC09 | Pending |
| CR-02 | RFID and external integration are out of scope due to budget | Project Scope Management | N/A | Confirmed |
| CR-03 | Must comply with data protection and accessibility standards | Legal Compliance Requirement | TC10 | Pending |

# **Tool Usage Explanation**

I have used Trello and GitHub tools for this project.

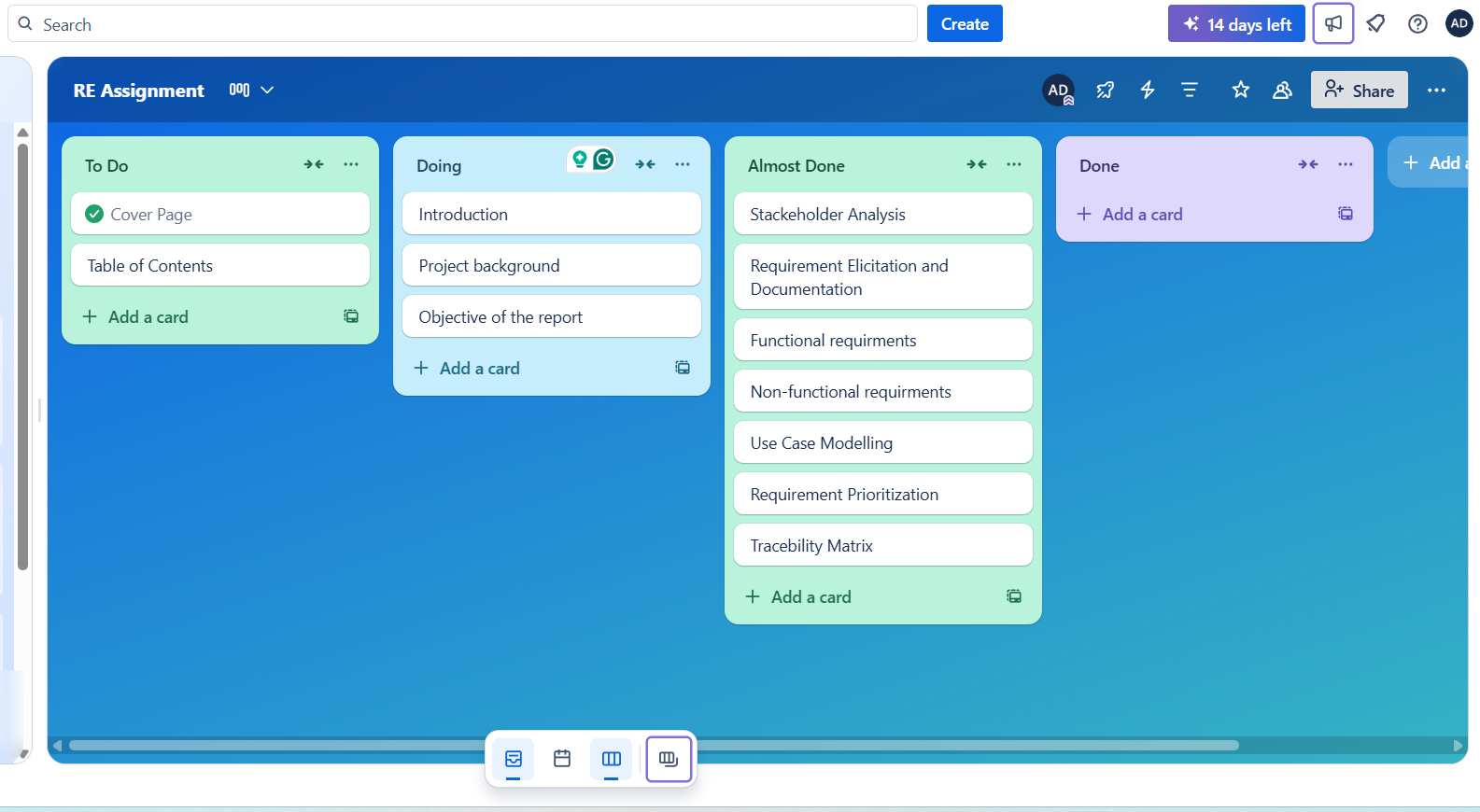
**Trello link**: <https://trello.com/invite/b/6839757ccdba6ad5c93476c9/ATTIe3fe032f82b002f9bc7a3bdd03045dad0FDD08FD/re-assignment>

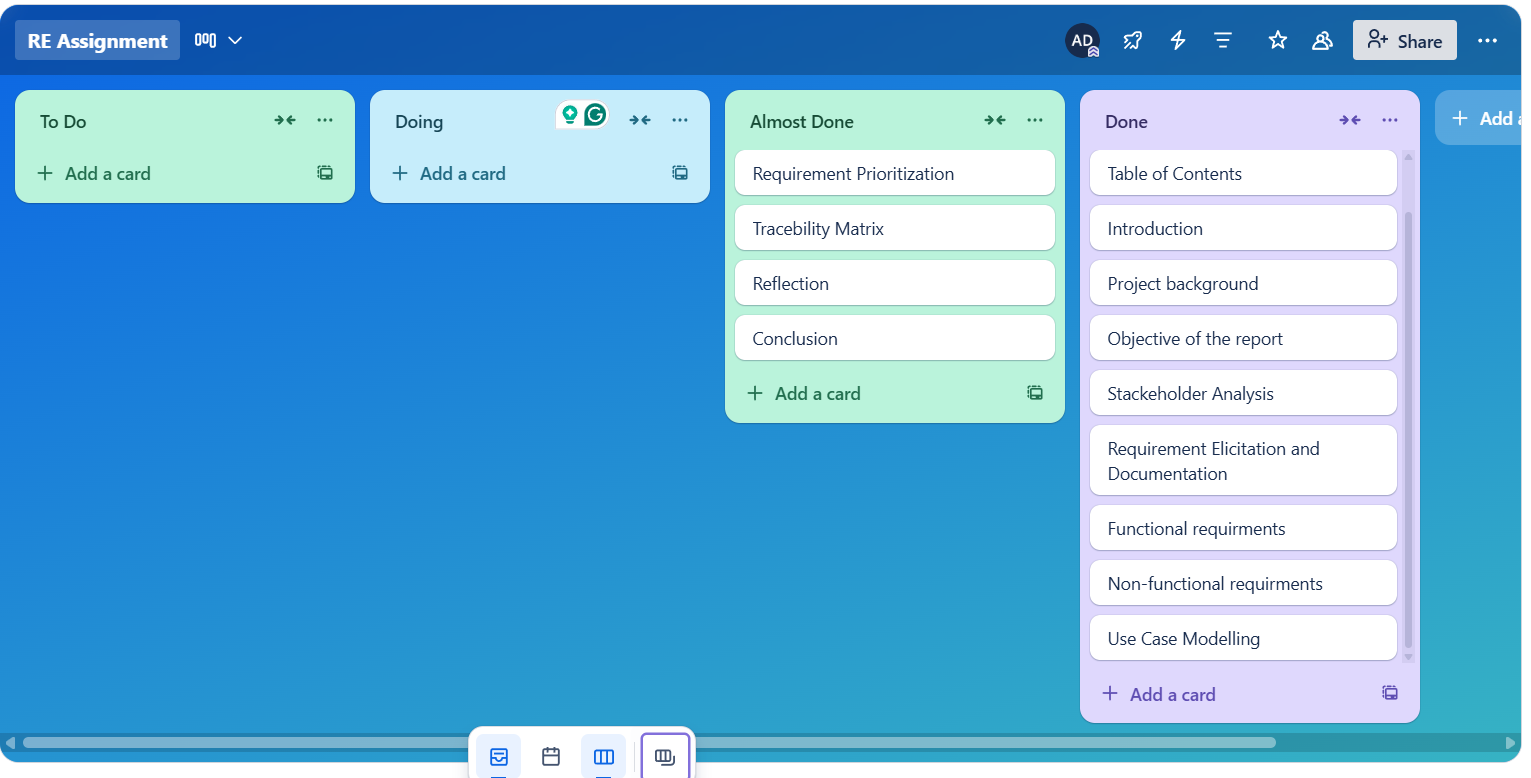
**GitHub link**:

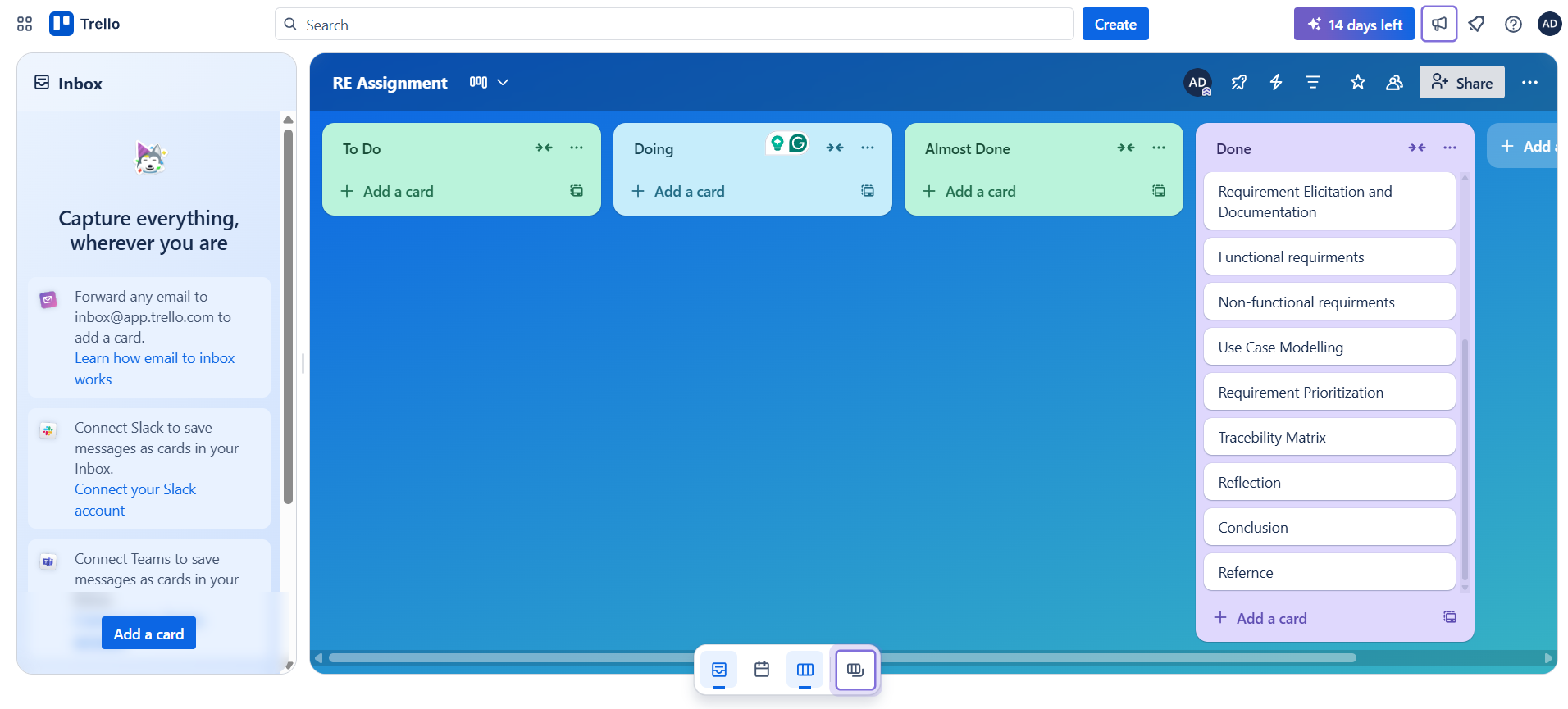
<https://github.com/alisenadanishwer/Requirement-Engineering-Assignment.git>

**Justification for Selection:**

Trello and GitHub are highly effective tools for project management. They facilitate efficient collaboration among employees or students by providing clear visibility of tasks and project progress. These tools enable users to manage their responsibilities promptly and systematically, ensuring improved productivity and organization.

**Screenshot of Trello:**   
  






**Reflection:**  
  
This project gave us some great takeaways about how to use requirements engineering in designing a system that actually works in the real world. While we were working on the Library Management System (LMS), we noticed a few important things:

* Engaging with Stakeholders is Key

Talking directly to different stakeholders helped us really understand what they needed and expected. This way, we made sure the requirements were relevant and focused on their real-life challenges.

* Benefits of a Structured Approach

By following a clear process for requirements engineering—from gathering to validating—we kept everything organized and clear. This helped avoid confusion during the design and development stages.

* Using Tools and Techniques

Incorporating tools like Trello and methods like MoSCoW prioritization and use case modeling made a big difference in managing tasks, improving communication, and keeping the requirements sorted throughout the project.

**Conclusion:**

In conclusion, this paper provides a systematic and successful way of designing a LMS based on fundamental principles of the requirements engineering. By involving stakeholders in the elicitation of system requirement the project was able to organize and structure and agenda and issues, the project was able to define both functional and non-functional system requirements properly in its requirement specification. Tools like Microsoft Word, Trello and GitHub worked well in establishing and monitoring tasks, and methods including MoSCoW prioritization and use case modelling provided clarity and focus to meet users’ needs. Constraints and system limitations were acknowledged with a sense of pragmatism, providing it contributed to a practical and achievable design. The report demonstrates a solid understanding of requirements engineering practices and lays the groundwork for an LMS that is efficient, scalable, and tailored to the needs of modern library environments.

# **Reference:**

Ali, S. (2017). Library Management System Mini Project Report on LIBRARY MANAGEMENT SYSTEM. *www.academia.edu*. <https://www.academia.edu/download/54051452/library_management_system.pdf>

Pressman, R. S., & Maxim, B. R. (2014). *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill Education.

Sommerville, I. (2011). *Software Engineering* (9th ed.). Addison-Wesley.

GeeksforGeeks. (2024, September 20). *Use case diagram for library management system*. GeeksforGeeks. <https://www.geeksforgeeks.org/use-case-diagram-for-library-management-system/>

Marthasari, G. I. (n.d.). *XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE Personal Extreme Programming with MoSCoW Prioritization for Developing Library Information System 1 st*.