REQUIREMENT SPECIFICATION DOCUMENT

For a Book tracking Application

*CS487 – Software Engineering*

The document provides an insight on understanding the requirements, scope and the context of a book tracking software application – Book-e-Tracker

TEAM:

- Praveen Kumar Selvan

- Vijay Bharathi

- Dayalini Nagaraj

- Parinitha Hirehal

1. **INTRODUCTION:**

The document describes the software requirements specifications for a book tracking software application called Book-e-Tracker. This tool/application mainly maintains a small repository of school books and tracks the books, classes, staff and students who check out the books

* 1. Goals and Objectives

Book-e-tracker is a standalone application built using Java to aid all its student users and staff users to manage their book checkouts and returns by simplifying the whole process.

* 1. Statement of Scope:

The Book-e-Tracker is an application which intends to automate the process of tracking the books. Basically it is a standalone application built using java to provide an interface for users to perform the operations. The system’s scope is mainly to provide the users/actors the ability to perform the following functionalities:

The student will be able to :

* Register for the application
* Login to the application
* Search for books
* Checkout the books
* Return the books
* Extend the dew date
* Logout

The staff will be able to:

* Register for the application
* Login to the application
* Search for the books
* Checkout the books
* Return the books
* Extend the dew date
* Authorise the book request
* Validate the book requests

The Admin will be able to:

* Provide different level of access to different users
* Maintain the book stack
* Add books
* Update the Books
* Delete books
* Add Users Information
* Delete Users Information
* Update Users Information
* Place order for the books
* Cancel orders
* Pay for the books
  1. Software Context

As mentioned earlier, the Book-e-Tracker is a standalone application providing the interface for book transactions.

Note: The information below is valid at this point of time. During the implementation, the design may vary or be enhanced.

The Application contains a LOGIN SCREEN prompting for a Student ID/ Staff ID and the password. The Login screen also contains the ‘register’ option for the first time users.

Based on the Kind of a login, the options appear in the next screen. Suppose if the user is a student, then a STUDENT SCREEN appears where s/he will be allowed to checkout and return books. If the user is a staff, STAFF HOME SCREEN appears, where s/he can authorize for the requests of books and validate the requests from students. And they can also checkout the books and return the books.

The CheckOut Screen will be used for displaying the item's title, description and selected quantity.

The admin has all the privileges open for him/her. The ADMIN HOME SCREEN have an option to add/update/delete the book information and the user information.

A PAYMENT SCREEN will be displayed for the admin to pay for the orders of books.

(Note: We will implement many more screen as we progress).

REQUIREMENTS :

The below table mentions the requirements to implement in our application Book-e-Tracker at a very high level.

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement No** | **Description** | **Priority** | **Tested** |
| 1 | The Repository of School books shall be maintained | High | No |
| 2 | The student shall check out Books (By Students-Limited by number of courses taken) | High | No |
| 3 | The staff shall check out Books (By Teacher- Limited to his/her department) | High | No |
| 4 | The Student/Staff shall return Books | High | No |
| 5 | The Admin Shall Add/Update/Delete the Book Information | High | No |
| 6 | The Admin shall Add/Update/Delete the Student Information | High | No |
| 7 | The Admin shall Add/Update/Delete the Course Information | High | No |
| 8 | ‘Search Book’ option for the application shall be enabled | Desired | No |
| 9 | The application shall track the Books | High | No |
| 10 | The application shall track the classes | High | No |
| 11 | The application shall track the students | High | No |
| 12 | The application shall track the total Number of books | High | No |
| 13 | The application shall track the total Number of books checked out | High | No |
| 14 | DataBase : MYSQL Server shall be implemented in the application | Desired | No |
| 15 | The application shall request for the ID and password | High | No |
| 16 | Security (Access based on users) aspects shall be implemented in the application | High | No |
| 17 | Response Time shall be implemented for 5 Seconds for the application | High | No |
| 18 | Reports ( List of all students who checked out books with due date and checked out date) shall be generated. | High | No |
| 19 | Reports( List of all authorized Users with access level) shall be generated. | High | No |
| 20 | Reports (List of all books in system with search feature) shall be generated. | High | No |
| 21 | Reports(List of all orders by order type) shall be generated. | High | No |
| 22 | Reports (List of all students and parents with due amount) shall be generated. | High | No |
| 23 | Reports (List of books checked out by Class/Book name/Book number) shall be generated. | High | No |
| 24 | Notifications shall be enabled | Desired | No |
| 25 | Autogenerate Fields : Book ID | High | No |
| 26 | Autogenerate Fields : Student ID | High | No |
| 27 | Autogenerate Fields : Staff ID | High | No |
| 28 | Student log details shall be recorded. | Desired | No |
| 29 | Time Stamp( For Check out and return) shall be recorded. | High | No |
| 30 | Pickup confirmation Number | High | No |
| 31 | Online Payment for due amount for admin shall be recorded. | Desired | No |
| 32 | Order Types shall be uploaded | High | No |
| 33 | Order Confirmation Number shall be enabled for the application | High | No |

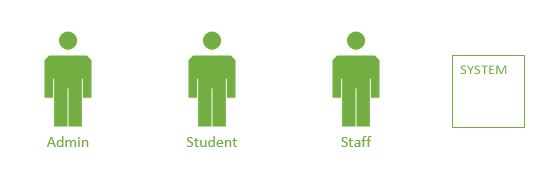
Requirements Table: (Functional and Non-fuctional Requirements)

1. **USAGE SENARIO**

This section mainly provides the usage scenario for the application Book-e-Tracker. The requirements mentioned in the above table are converted as use cases.

* 1. User Profiles/Actors

The important actors here are : Admin, Student, Staff and the System



* 1. Use-Cases

Some of the important uses cases are listed below:

* Login
* Logout
* Register
* Request for a Book
* Checkout Book
* Return Book
* Extend the dew date
* Maintain a book stack
* Add Book
* Update Book
* Delete Book
* Order for books
* Cancel orders
* Maintain User information
* Add user Information
* Delete user Information
* Update user Information
* Add course information
* Update course information
* Delete course information
  1. USE-CASE DIAGRAMS

1. Admin – System Use case scenario:



1. Staff – SystemUse Case Senario :



3. Student – System Use Case Senario :



**3.0 DATA MODEL AND DESCRIPTION**

The book tracking application encompasses the following information domains:

i)  Books Warehouse

ii) Users

iii) Courses

           iv) Orders

**3.1 Data Description:**

* **Books:** A book entity with information like name, ISBN and Id
* **Users:** Any person with a valid username and password who can login to the system
* **Student:** A student is a user who can enroll in courses and borrow books
* **Teacher:** A staff is a user who teaches courses and borrows books
* **Administrator:** Admins areusers who can place order for the library
* **Courses:** A course entity representing the course name and ID
* **Order :** Order is an entity describing orders for the library placed by admin

**Association Tables**

* **Placed By Table:** Describes a relationship between admin and order
* **Borrowed By Table:** Describes relationship between user and book
* **Enrolls In Table:** Describes relationship between student and course
* **Teaches Table:** Describes relationship between teacher and course

**3.2 Data Objects :**

1. **Books Table :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **DataType** | **Size** | **Nullable** | **Constraint** |
| School\_Book\_ID | Integer | 15 | No | Primary Key |
| Book\_Name | Varchar | 30 | No |  |
| ISBN | Integer | 16 | No |  |

**2.  Users Table :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FieldName | DataType | Size | Nullable | Constraint |
| User\_Name | Varchar | 15 | No | PrimaryKey |
| Password | Varchar | 15 | No |  |
| Name | Varchar | 15 | No |  |
| Access\_level | Varchar | 10 | No |  |

**3.  Student Table :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FieldName | DataType | Size | Nullable | Constraint |
| Student\_Id | Integer | 10 | No | Primary Key |
| Parent\_Name | Varchar | 20 |  |  |

**4. Teachers Table :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FieldName | DataType | Size | Nullable | Constraint |
| Staff\_Id | Integer | 10 | No | Primary Key |

**5. Administrator Table :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FieldName | DataType | Size | Nullable | Constraint |
| Admin\_Id | Integer | 10 | No | Primary Key |

**6.  Courses Table :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FieldName | DataType | Size | Nullable | Constraint |
| Course\_Id | Integer | 10 | No | Primary Key |
| Course\_Name | Varchar | 20 |  |  |

**7. Order Table :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FieldName | DataType | Size | Nullable | Constraint |
| Order\_Id | Integer | 10 | No | Primary Key |
| Order\_Type | Varchar | 20 |  |  |
| Order\_Description | Varchar | 20 |  |  |

**Association Tables :**

**1. Placed By Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FieldName | DataType | Size | Nullable | Constraint |
| Order\_Id | Integer | 10 | No | Foreign key on orders table  Part of Primary Key |
| Admin\_Id | Integer | 10 | No | Foreign key on admin table  Part of primary key |
| Placed\_Date | Date |  |  |  |
| Order\_Status | Varchar | 20 |  |  |

**2.  Borrowed By Table :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FieldName | DataType | Size | Nullable | Constraint |
| User\_Name | varchar | 15 | No | Foreign Key on User Table  Part of Primary Key |
| School\_Book\_ID | Integer | 15 | No | Foreign Key on Books Table  Part of Primary Key |
| Amount\_Due | Integer |  |  |  |
| Due\_Date | Date |  |  |  |
| Return\_Date | Date |  |  |  |
| Checkout\_Date | Date |  |  |  |

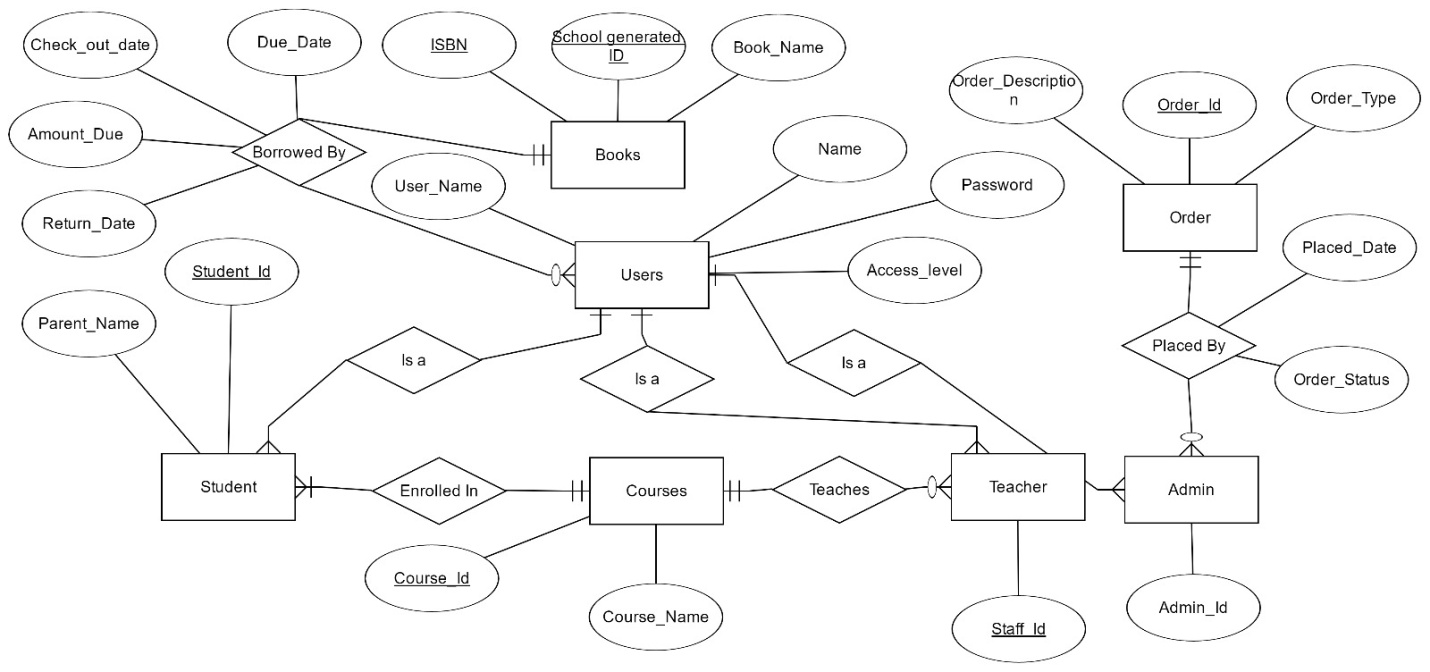
**3. Enrolls In Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FieldName | DataType | Size | Nullable | Constraint |
| Course\_Id | Integer | 10 | No | Foreign Key on Courses  Table  Part of Primary Key |
| Student\_Id | Integer | 10 | No | Foreign Key on Students Table  Part of Primary Key |

**4.  Teaches Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FieldName | DataType | Size | Nullable | Constraint |
| Course\_Id | Integer | 10 | No | Foreign Key on Courses  Table  Part of Primary Key |
| Staff\_Id | Integer | 10 | No | Foreign Key on Staff Table  Part of Primary Key |

3.3 Entity Relationship Diagram



**4.0 FUNCTIONAL MODEL AND DESCRIPTION**

This section describes the static structure of the software.

**4.1 Class diagrams**

**4.2 Software Interface Description**

The software interface(s)to the outside world is(are) described.

**4.2.1 External machine interfaces**

Interfaces to other machines (computers or devices) are described.

As all computing will be done in a single system, no external machine interface will be needed

**4.2.2 External system interfaces**

Interfaces to other systems, products or networks are described.

The Application will interact with MySQL database server.

**4.2.3 Human interface**

An overview of any human interfaces to be designed for the software is presented.

Users have appropriate privileges. Based on the privileges appropriate operations can be performed by the user and the admin through Graphical User Interface coded in Java.

**5.0 BEHAVIORAL MODEL AND DESCRIPTION**

A description of the behavior of the software is presented.

**5.1 Description for software behavior**

A detailed description of major events and states is presented in this section.

**5.1.1 Events**

A listing of events (control, items) that will cause behavioral change within the system is presented.The consequence of events will be as follows: The user will need to log on to the system either as a Student, Teacher or Admin.

User (Student, Teacher) will be directed to the default Home screen which has the following options View available/search Books, My Orders, My Books Tab.

Home Screen (Search Book): A list of available books and an option to search will be displayed. User can select a books and click on “Request Book” button to request a book.

My Orders: On request, book is added to ‘My Orders’ Tab where user can track the status of his request.

My Books: Displays all the books the user has checked out with due date and an option to return book.

Admin: Admin will be directed to the Management Screen where they will be able to add a new user, accept/deny Orders (Book request), add/edit book details.

In general, the system will respond to the user’s actions according to their access level. All User Screens will have Home, Search Books, My Orders, My Books and Help tabs which will direct the user to the respective screen.

Management Screen will contain reports on list of authorized users and access level, list of books, list of orders and list of books checked out.

**5.1.2 States**

A listing of states (modes of behavior) that will result as a consequence of events is presented.

The software will have the following modes of behavior:

Login Page: Accept input from the user

Main Menu: List options available to user

View Books: Retrieve information from the database and display it to the user

Borrowed Books: Lists books borrowed by user

Request Book: Make a request for the book user requires

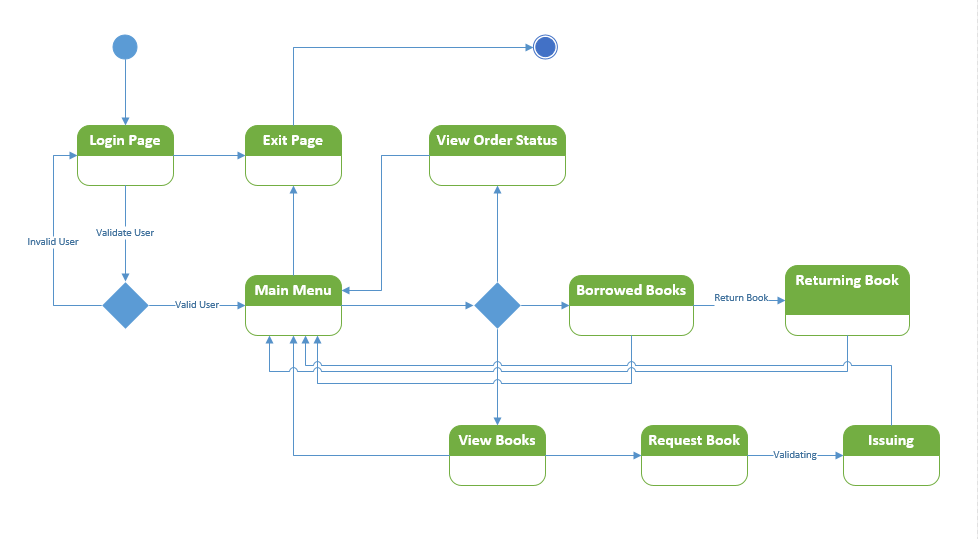
Issue: Validate user and issue book to the user

Return Book: Return the book to the library.

View Order Status: View status of orders placed

**5.2 State Chart Diagram**

Depict the overall behavior of the system.



**6.0 RESTRICTIONS, LIMITATIONS, AND CONSTRAINTS**

The below mentioned Restrictions, Limitations, and Constraints explains the cases which impact the specification, design, or implementation of the software system. These items may also contribute to restrict the scalability and performance of the system as well.

**Constraints:**

* The users must have their correct usernames and passwords to enter into the Application.

**Assumptions and dependencies**

* The users have sufficient knowledge of computers.
* The computer should have the product installed correctly.
* The users know the English language, as the user interface will be provided in English

**7.0 VALIDATION CRITERIA**

Validation or Evaluation of results is one of the most important parts of the project and the findings are reviewed to reflect the status. This step promotes the project heading in the correct direction and reaching a consensus.

**7.0.1Software Testing**

Software testing is an important element of software utility assurance and represents the ultimate review of specification, design and coding. The increasing visibility of the software as a system element and the costs associated with project failure are motivating forces for well planned, through testing.

Though the test phases is often thought of as separate and distinct from the development effort first develop and then test, Testing is a concurrent process that provides valuable information for the development team.

Through Software validation we can get the information about the quality of the product or service under test. It will provide an objective and independent view of the software and to understand the risk involved. Test techniques include the process of executing a program or application with the intent of finding [software bugs](https://en.wikipedia.org/wiki/Software_bug), errors or other defects and to verify that the software product is fit for use.

Software testing involves the execution of a software component or to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

* Meets the client requirements and design
* Responds correctly to all types of input
* Function performance and time taken to execute it.
* User friendly
* Can be installed and run in its intended [environments](https://en.wikipedia.org/wiki/Operating_environment), and
* Achieves the general result its stakeholders desire.

There are several rules that can serve as testing objective. They are

1. Testing is a process of executing a program with the intent of finding an error.
2. A good test case is one that has a high profitability of finding an undiscovered error.
3. A successful test is one that uncovers an undiscovered error.

If testing is conducted successfully according to the objectives stated above, it will uncover errors in the software.

**7.1 Classes of tests**

Testing is the process of executing the program with the intent of finding errors. Testing cannot show the absence of defects, it can only show the errors that are present and helps the developer to validate the software. The testing principles used are,

* Tests are traceable to customer requirements.
* 80% of errors will likely be traceable to 20% of program modules.
* Testing should begin in-small and progress towards testing in-large.

Types of tests to be conducted:

* Unit Testing
* Integration Testing
* Functional Testing
* System Testing
* White Box Testing
* Black Box Testing
* Performance Testing

Component Interface Testing, Operational Acceptance Testing, Smoke and sanity testing, Regression Testing, Testing based on data flow , Scenarios, UI, Client Server Architecture testing are other testing techniques which are useful for software Validation.

**Unit Testing:**

Unit testing is a software development process in which the small parts called units of an application, are individually tested by the developer. The update, delete, view , UI elements are tested by the developers.

**Functional Testing:**

The functionality and the features of the application are tested in this phase. All the features mentioned in the requirement has to be implemented and validated. According to our application, add, delete, update, search, view, order, payment, login and logout are the main functionalities that has to be tested for validation.

**White Box Testing:**

Test the software based on the internal structures and working of the application, as opposed to its functionality.

**Black Box Testing:**

The whole application is tested based on GUI, Design, Features and flow of the application and validated based on that.

**7.2 Expected software response**

Based on each scenarios and modules, test cases will be generated and analysis will be done. If we get expected results, it means that we are good and the requirements are satisfied. If the actual results are different from the expected then we need to go through the software code and fix the error.

It also includes UI specifications, Data to be displayed and the performance of the software based on the response time.

**Function and Expected Results:**

* 1. **Log In and Logout**

**Case1**: If the user is already a member then the user should be able to login using the username and password.

**Case 2** : If the user is not a member, then he/she has to create a new account by clicking “New User” button.

**Case 3:** On click on ‘New User’ button, the user should be taken to a page with a form.

**Case 4:** The new user should be able to fill all the required information and able to create a new account.

**Case 5 :** Based on the user information the users access level must be decided and the access are provided based on that.

**Case 6:** If the user is logged in then the user must be able to log out.

* 1. **Welcome Screen**

**Case1:** The details of the user has to be displayed in the screen and other information related to the user.

**Case 2:** Search bar must be available.

**Case 3:** Other UI elements like necessary buttons and texts should be displayed and align according to the design.

* 1. **Add/ Update/Delete**

**Case1:** Students must be able to add their details, update their information and delete any unnecessary information.

**Case2**: Student must be able to fill all the required information like name, student id and courses registered. If the required information is not filled then “error message” should be displayed.

**Case 3**: Teachers must be able to fill their information and courses handled by them. If the required details are not filled then error message should be displayed.

**Case 4:** Admin must be able to add the books to the database, availability along with course id, author and book name.

**Case5**:The users should be able to modify the information or delete the information.

* 1. **Search**

**Case 1**: The users should be able to search the books based on the title, keywords, author, course name.

**Case2**: The results of the search must be displayed to the user.

**Case3**: The user must be able to clear the search contents and search for other books.

**Case4:** The user must be able to select the book from the search contents.

* 1. **Order**

**Case1:** The UI should align with the design

**Case2:** The user must be able to select the book and proceed to order it by clicking order button.

**Case3:** The user must be able to order the book based on the availability and the course registered for. If the user does not satisfy the conditions then the appropriate error message should be displayed.

**Case4:** The user should be able to order the book and notified once the order has been confirmed.

* 1. **Payment**

**Case1:** The user should be able to pay the amount due for the book.

**Case2:** If the payment is successful, the user should be notified with success message else appropriate error message should be displayed.

**Case3:** The balance amount due should be updated in the user profile.

* 1. **Order Approval**

**Case1:** The Admin should be able to view all the pending approvals.

**Case2:** The Admin should be able to approve the order based on certain conditions

.

**Case3:** The Admin should be able to add book availability, the checkout date and the return date based on the user profile.

* 1. **Return book.**

**Case1**: The Admin should be able to update the status of the book and the user’s status based on the return date.

**Case2**: The user should be able to view the full details of the order along with borrow, payment and return date and the status of the order once the admin has updated it.

**7.3 Performance bounds**

**1. Response Time**

The time taken to create account , add, delete and update information should not take more time. The amount of time required to retrieve the details from the database should be less. The response message to the user or to the admin should be delivered immediately and there should not be any delay. When large amount of users are using the system at same time, then the process can take more time and the user must be notified with proper message.

**2. Capacity**

The system should be capable of handling large number of users at the same time. The database should be accessed by multiple users at the same time and the database should be updated accordingly. Internal errors should not occur and it should not be crashed.

**3. Resource Availability and Utilization**

The books available should be updated by the admin frequently according to the orders. The users will be able to access it and order only based on the availability.

**4. Browser Compatibility**

This software will run only on web browsers based on their versions and the Internet connection is mandatory.