SOFTWARE REQUIREMENTS SPECIFICATION

For

Blog Platform

Prepared by :-

Pravin Kumar R
Mohanasundaram S
Vigneshwaran G

1. Introduction

1.1 Purpose

The main objective of this document is to illustrate the requirements of the project Blog Platform. The document gives the detailed description of the both functional and non-functional requirements proposed by the client. The purpose of this project is to provide a friendly environment to provide online presence and effectively sharing their stories of aspiring writers, content creators and bloggers. The main purpose of this project is to provide an enriched experience for both bloggers and readers. This project describes the software interface requirements using ER diagrams and UML diagrams.

1.2 Document Conventions

- Entire document should be justified.
  - Convention for Main title
    - Font face: Times New Roman
    - Font style: Bold
    - Font Size: 14
  - Convention for Sub title
    - Font face: Times New Roman
    - Font style: Bold
    - Font Size: 12
  - Convention for body
    - Font face: Times New Roman
    - Font Size: 12

1.3 Scope of Development Project

Blogging is a great way to share your thoughts and ideas with the world, build an audience, and even make money. Blogging platforms are broadcast-style communications systems that enable authors to publish articles, opinions or product reviews (known as posts), which can be delivered through stand-alone websites, email, feed syndications systems and social networks. People will likely be interested in content forms that can be easily consumed. The project is specifically designed for bloggers and readers.

The product will work as a complete user interface for the blog platform must support a variety of content forms, including traditional text-based posts as well as emerging formats such as videos, vlogs, and other visual content. A user-friendly interface is paramount, ensuring both bloggers and readers can navigate the platform effortlessly. Responsive design across devices, intuitive content discovery, and easy-to-use editing tools contribute to an enhanced user experience.

The project can be easily implemented under various situations. We can add new features as and when we require, making reusability possible as there is flexibility in all the modules. The language used for developing the project is Java as it is quite advantageous than other languages

in terms of performance, tools available, cross platform compatibility, libraries, cost (freely available), and development process.

1.4 Definitions, Acronyms and Abbreviations

JAVA -> platform independence
SQL -> Structured query Language
ER -> Entity Relationship
UML -> Unified Modelling Language
IDE -> Integrated Development Environment
SRS -> Software Requirement Specification

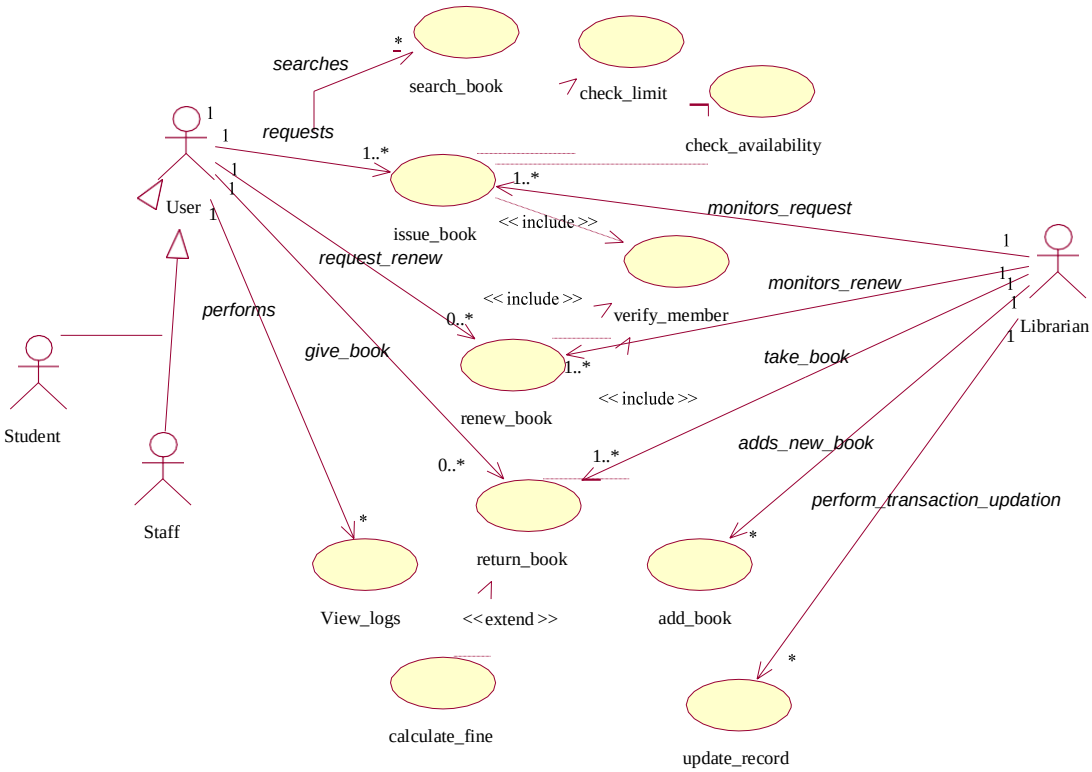
1.5 References

- Books
  - Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices (ACM Press) by Michael Jackson
  - Software Requirements (Microsoft) Second Edition By Karl E. Wiegers
  - Software Engineering: A Practitioner's Approach Fifth Edition By Roger S. Pressman
- Websites **http://www.slideshare.net/**
- **https://www.blogger.com/blog/posts/5452712659477338226**

2. Overall Descriptions

2.1 Product Perspective

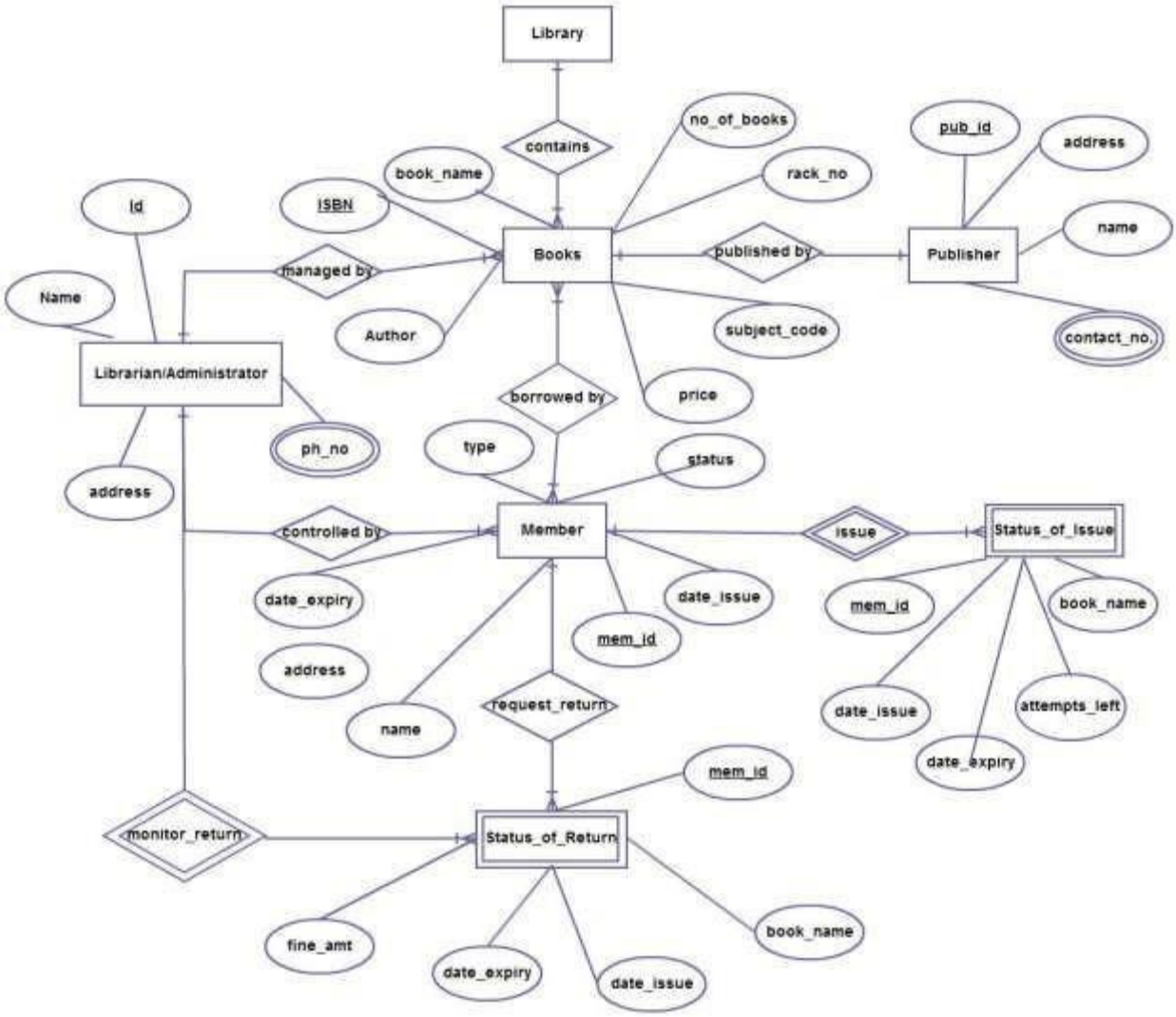
Use Case Diagram of Blog Platform



This is a broad level diagram of the project showing a basic overview. The users can be either staff or student. This System will provide a search functionality to facilitate the search of resources. This search will be based on various categories viz. book name or the ISBN. Further the library staff personnel can add/update the resources and the resource users from the system. The users of the system can request issue/renew/return of books for which they would have to follow certain criteria.

2.2 Product Function

Entity Relationship Diagram of Blog Platform



The Online Library System provides online real time information about the books available in the Library and the user information. The main purpose of this project is to reduce the manual work. This software is capable of managing Book Issues, Returns, Calculating/Managing Fine, Generating various Reports for Record-Keeping according to end user requirements. The Librarian will act as the administrator to control members and manage books. The member's status of issue/return is maintained in the library database. The member's details can be fetched by the librarian from the database as and when required. The valid members are also allowed to view their account information.

2.3 User Classes and Characteristics

The system provides different types of services based on the type of users [Member/Librarian]. The Librarian will be acting as the controller and he will have all the privileges of an administrator. The member can be either a student or staff of the university who will be accessing the Library online.

The features that are available to the Librarian are:- ➢

A librarian can issue a book to the member.

- Can view the different categories of books available in the Library
- Can view the List of books available in each category
- Can take the book returned from students
- Add books and their information to the database
- Edit the information of existing books
- Can check the report of the existing books
- Can check the report of the issued books
- Can access all the accounts of the students

The features that are available to the Members are:-

- Can view the different categories of books available in the Library ➢ Can view the List of books available in each category ➢ Can own an account in the library.
- Can view the books issued to him
- Can put a request for a new book
- Can view the history of books issued to him previously
- Can search for a particular book

2.4 Operating Environment

The product will be operating in windows environment. The blog platform is a website and shall operate in all famous browsers, for a model we are taking Microsoft Internet Explorer, Google Chrome and Mozilla Firefox. Also it will be compatible with the IE 6.0. Most of the features will be compatible with the Mozilla Firefox & Opera 7.0 or higher version. The only requirement to use this online product would be the internet connection.

The hardware configuration include Hard Disk: 40 GB, Monitor: 15" Color monitor, Keyboard: 122 keys. The basic input devices required are keyboard, mouse and output devices are monitor, printer etc.

2.5 Assumptions and Dependencies

The assumptions are:-

- The coding should be error free
- The system should be user-friendly so that it is easy to use for the users
- The information of all users, books and libraries must be stored in a database that is accessible by the website
- The system should have more storage capacity and provide fast access to the database
- The system should provide search facility and support quick transactions
- The Library System is running 24 hours a day
- Users may access from any computer that has Internet browsing capabilities and an Internet connection
- Users must have their correct usernames and passwords to enter into their online accounts and do actions

The dependencies are:-

- The specific hardware and software due to which the product will be run
- On the basis of listing requirements and specification the project will be developed and run
- The end users (admin) should have proper understanding of the product
- The system should have the general report stored
- The information of all the users must be stored in a database that is accessible by the Library System
- Any update regarding the book from the library is to be recorded to the database and the data entered should be correct

2.6 Requirement

Software Configuration:-
This software package is developed using java as front end which is supported by sun micro system. Microsoft SQL Server as the back end to store the database. Operating System: Windows NT, windows 98, Windows XP
Language: Java Runtime Environment, Net beans 7.0.1 (front end)
Database: MS SQL Server (back end)

Hardware Configuration:-
Processor: Pentium(R)Dual-core CPU
Hard Disk: 40GB
RAM: 256 MB or more

2.7 Data Requirement

Data Requirements includes user information (profiles and roles), blog post details (title, content, author), media management, user interactions (comments, likes, shares), security measures (password policies, authentication), search and navigation features, a notification system, analytics, responsive design considerations, performance metrics, content management tools (drafts, revisions), localization, third-party integrations, legal and compliance guidelines, backup and recovery plans, API documentation, testing criteria, deployment specifics, and support/maintenance details. Constraints and assumptions are also addressed. This comprehensive document serves as a guide for the development, deployment, and maintenance of our blog platform.

3. External Interface Requirement

3.1 GUI

The blog platform boasts an intuitive graphical interface tailored for both users and administrators. Admins can seamlessly execute tasks such as creating, updating, and viewing blog details. Noteworthy features include:

- Quick Reports: Users can swiftly generate reports on blog activity within specific time frames.
- Search Functionality: The platform facilitates searches based on diverse criteria.
- Customizable UI: Administrators enjoy the flexibility to customize the user interface.

- Modular Integration: All software modules seamlessly integrate into the graphical user interface, adhering to established standards.
- Standardized Design: The design is user-friendly and maintains a consistent template.
- User Management Integration: The interface effortlessly interacts with the user management module, incorporating a dedicated section for the login/logout module.

Login Interface:-
The login interface ensures a secure and user-friendly experience. New users can register by entering their details to create an account. Subsequent logins require the user to input their username and password, with an error message prompt for incorrect entries.

Search:-
The search functionality empowers users to find specific blog posts by type or title.

Categories View:-
The Categories view displays available blog categories and provides administrators the ability to add, edit, or delete categories.

Administrator's Dashboard:-
This control panel empowers administrators to manage users, blog posts, and various settings. Admins can add or remove users, edit blog posts, and control other functionalities.

4. System Features

The users of the system should be provided the surety that their account is secure. This is possible by providing:-

- User authentication and validation of members using their unique member ID
- Proper
- Proper accountability which includes not allowing a user to see other user's account. Only administrator will see and manage all member accounts

5. Other Non-functional Requirements

5.1 Performance Requirement

The blog platform must exhibit superior performance to meet the dynamic needs of its users within the anticipated university environment. It should ensure swift and accurate execution of functions, providing a seamless experience for both content creators and readers. The system should handle large volumes of diverse content, including blog posts, comments, and media, without compromising speed or responsiveness. Robust error handling mechanisms must be in place to promptly identify and address issues, preventing data loss and minimizing downtime. The platform's scalability is crucial, enabling it to accommodate a growing user base and an expanding repository of content.

- The performance of the website should be fast and accurate

- This website prevent loss in information and long downtime period. Thus it should have inbuilt error testing to identify invalid username/password

5.2 Safety Requirement

Ensuring the safety and security of a blog platform is paramount in safeguarding user data and maintaining a trustworthy environment. Implementation of robust encryption protocols is crucial to protect sensitive information such as user credentials and personal details during transmission and storage. Regular security audits and vulnerability assessments help identify and address potential weaknesses in the system. Secure user authentication mechanisms, such as multi-factor authentication, bolster account protection. Additionally, incorporating effective user authorization controls restricts access to sensitive features and data.

5.3 Security Requirement

- System will use secured database
- Normal users can just read information but they cannot edit or modify anything except their personal and some other information.
- System will have different types of users and every user has access constraints
- Proper user authentication should be provided
- No one should be able to hack users' password
- There should be separate accounts for admin and members such that no member can access the database and only admin has the rights to update the database.

5.4 Requirement attributes

- There may be multiple admins creating the project, all of them will have the right to create changes to the system. But the members or other users cannot do changes
- The project should be open source
- The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database
- The user be able to easily to write their contents with the system.

5.5 Business Rules

A business rule is anything that captures and implements business policies and practices. A rule can enforce business policy, make a decision, or infer new data from existing data. This includes the rules and regulations that the System users should abide by. This includes the cost of the project and the discount offers provided. The users should avoid illegal rules and protocols. Neither admin nor member should cross the rules and regulations.

5.6 User Requirement

In designing a comprehensive blog platform to cater to the needs of both members and administrators, the primary focus is on creating an intuitive and user-friendly interface. Members, with basic computer and internet knowledge, should find it easy to navigate through blog creation, editing, and commenting features. Concurrently, administrators, possessing advanced system knowledge, are tasked with system maintenance and issue resolution. A well-documented user manual, supplemented by online help and installation guides, ensures users at all levels can

seamlessly navigate the platform. Essential administrator facilities, including backup and recovery, password retrieval, data migration, replication, auto-recovery, and file organization, are integrated to enhance system reliability and user experience. Moreover, security measures, scalability considerations, and a commitment to regular server maintenance and updates are embedded to guarantee optimal performance and robustness. The proper user interface, user manual, online help and the guide to install and maintain the system must be sufficient to educate the users on how to use the system without any problems.

The admin provides certain facilities to the users in the form of :-

- Backup and Recovery
- Forgot Password
- Data migration i.e. whenever user registers for the first time then the data is stored in the server
- Data replication i.e. if the data is lost in one branch, it is still stored with the server
- Auto Recovery i.e. frequently auto saving the information
- Maintaining files i.e. File Organization
- The server must be maintained regularly and it has to be updated from time to time

6. Other Requirements

6.1 Data and Category Requirement

A comprehensive blog platform necessitates the management of diverse datasets across various categories. User-centric data forms a foundational element, encompassing user profiles with essential details such as usernames, emails, and securely hashed passwords. Preferences, including theme choices and notification settings, contribute to a personalized user experience. Authentication tokens play a pivotal role in ensuring secure user sessions. Content data comprises blog posts with associated metadata, including titles, bodies, authors, publication dates, and tags for effective categorization. Notification systems, both email and in-app, keep users informed about new posts and interactions. Accessibility considerations ensure inclusivity, adhering to standards that cater to individuals with disabilities. Developing a successful blog platform requires a holistic approach, addressing frontend and backend intricacies while prioritizing security and user experience. Depending upon the category of user the access rights are decided. It means if the user is an administrator then he can be able to modify the data, delete, append etc. All other users except the Similarly there will be different categories of blogs are available. According to the categories of blog their relevant data should be displayed. The categories and the data related to each category should be coded in the particular format.

6.2 Appendix

A: Admin, Abbreviation, Acronym, Assumptions; B: Books, Business rules; C: Class, Client, Conventions; D: Data requirement, Dependencies; G: GUI; K: Key; L: Library, Librarian; M: Member; N: Non-functional Requirement; O: Operating environment; P: Performance, Perspective, Purpose; R: Requirement, Requirement attributes; S: Safety, Scope, Security, System features; U: User, User class and characteristics, User requirement;

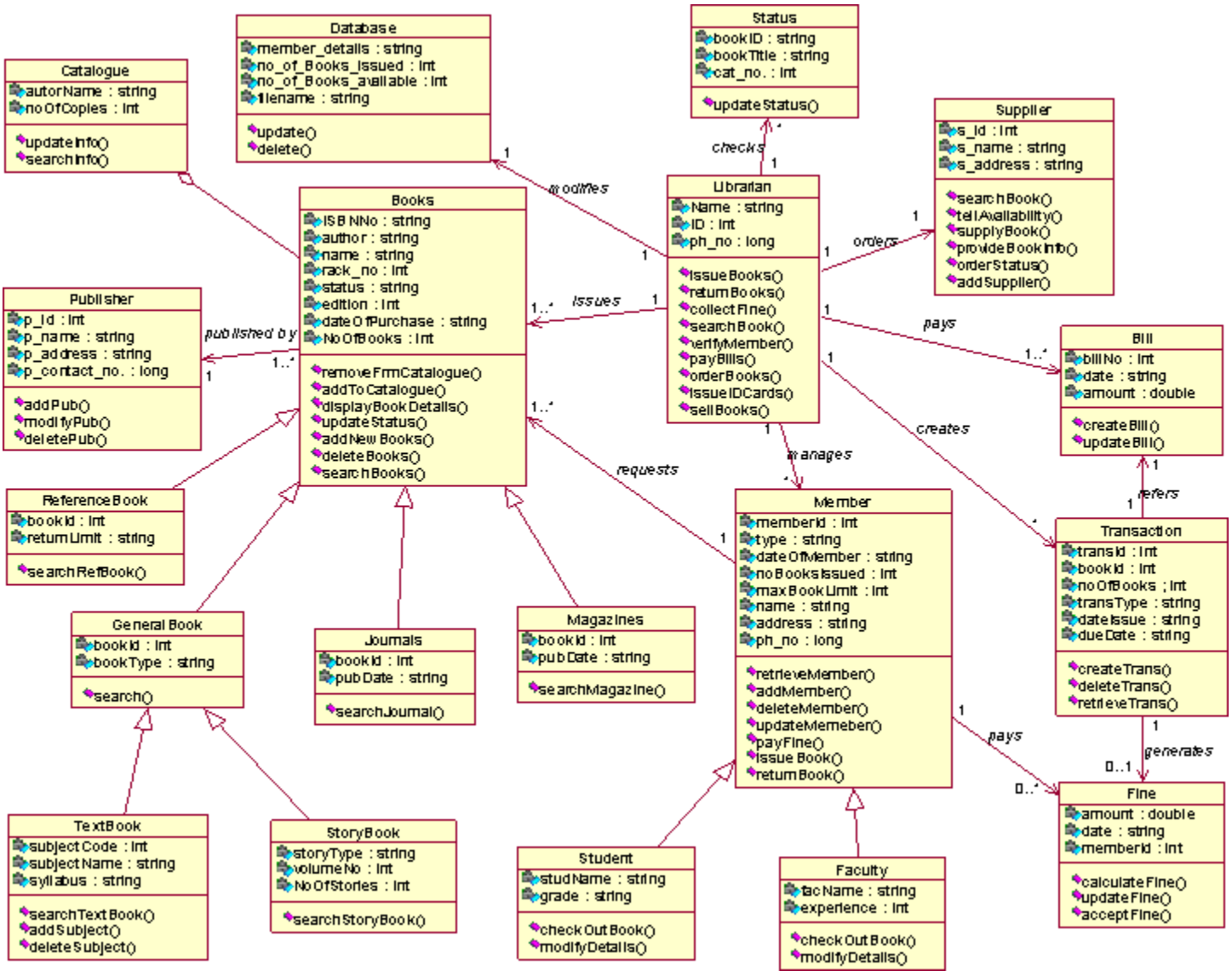
6.3 Glossary

The following are the list of conventions and acronyms used in this document and the project as well:

- Administrator: A login id representing a user with user administration privileges to the software
- User: A general login id assigned to most users
- Client: Intended users for the software
- SQL: Structured Query Language; used to retrieve information from a database
- SQL Server: A server used to store data in an organized format
- Layer: Represents a section of the project
- User Interface Layer: The section of the assignment referring to what the user interacts with directly
- Application Logic Layer: The section of the assignment referring to the Web Server. This is where all computations are completed
- Data Storage Layer: The section of the assignment referring to where all data is recorded
- Use Case: A broad level diagram of the project showing a basic overview
- Class diagram: It is a type of static structure diagram that describes the structure of a system by showing the system's cases, their attributes, and the relationships between the classes
- Interface: Something used to communicate across different mediums
- Unique Key: Used to differentiate entries in a database

6.4 Class Diagram

A class is an abstract, user-defined description of a type of data. It identifies the attributes of the data and the operations that can be performed on instances (i.e. objects) of the data. A class of data has a name, a set of attributes that describes its characteristics, and a set of operations that can be performed on the objects of that class. The classes' structure and their relationships to each other frozen in time represent the static model. In this project there are certain main classes which are related to other classes required for their working. There are different kinds of relationships between the classes as shown in the diagram like normal association, aggregation, and generalization. The relationships are depicted using a role name and multiplicities. Here 'Blogger', 'Reader' and 'Blog' are the most important classes which are related to other classes.


Database
member_details : string
no_of_Books_Issued : int
no_of_Books_available : int
filename : string
update()
delete()
Status
bookID : string
bookTitle : string
cat_no. : int
updateStatus()
Catalogue
autorName : string
noOfCopies : int
updateInfo()
searchInfo()
Supplier
s_id : int
s_name : string
s_address : string
searchBook()
tellAvailability()
supplyBook()
provideBookInfo()
orderStatus()
addSupplier()
checks
modifies
Librarian
Name : string
ID : int
ph_no : long
issueBooks()
returnBooks()
collectFine()
searchBook()
verifyMember()
payBills()
orderBooks()
issueIDCards()
sellBooks()
orders
Books
ISBNNo : string
author : string
name : string
rack_no : int
status : string
edition : int
dateOfPurchase : string
NoOfBooks : int
removeFrmCatalogue()
addToCatalogue()
displayBookDetails()
updateStatus()
addNewBooks()
deleteBooks()
searchBooks()
issues
Publisher
p_id : int
p_name : string
p_address : string
p_contact_no. : long
addPub()
modifyPub()
deletePub()
published by
pays
Bill
billNo : int
date : string
amount : double
createBill()
updateBill()
creates
refers
manages
requests
ReferenceBook
bookId : int
returnLimit : string
searchRefBook()
Member
memberId : int
type : string
dateOfMember : string
noBooksIssued : int
maxBookLimit : int
name : string
address : string
ph_no : long
retrieveMember()
addMember()
deleteMember()
updateMemeber()
payFine()
Issue Book()
returnBook()
Transaction
transId : int
bookId : int
noOfBooks : int
transType : string
dateIssue : string
dueDate : string
createTrans()
deleteTrans()
retrieveTrans()
generates
pays
GeneralBook
bookId : int
bookType : string
search()
Journals
bookId : int
pubDate : string
searchJournal()
Magazines
bookId : int
pubDate : string
searchMagazine()
Fine
amount : double
date : string
memberId : int
calculateFine()
updateFine()
acceptFine()
TextBook
subjectCode : int
subjectName : string
syllabus : string
searchTextBook()
addSubject()
deleteSubject()
StoryBook
storyType : string
volumeNo : int
NoOfStories : int
searchStoryBook()
Student
studName : string
grade : string
checkOutBook()
modifyDetails()
Faculty
facName : string
experience : int
checkOutBook()
modifyDetails()