

BOOK BANK MANAGEMENT

18CSC209J - Database Management System and Cloud Integration Services

Mini Project Report

Submitted by

**Sumith Sai K [RA2111028010164]
B.Tech. CSE - Cloud Computing**

**Simran Pradhan [RA2111028010190]
B.Tech. CSE - Cloud Computing**

**Naresh K [RA2111028010158]
B.Tech. CSE - Cloud Computing**



**DEPARTMENT OF NETWORKING AND COMMUNICATION
SCHOOL OF COMPUTING
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

(Under Section 3 of UGC Act, 1956)

**S.R.M. NAGAR, KATTANKULATHUR – 603 203
KANCHEEPURAM DISTRICT**

MAY 2023

TABLE OF CONTENTS

Chapter No.	Title	Page No.
1.	Abstract	
2.	Introduction	
3.	Problem Statement	
4.	Module Description	
5.	Use case diagram	
6.	ER diagram	
7.	Database Creation using DDL and DML	
8.	Normalization of Database	
9.	Implementation using Dynamo DB	
10.	Conclusion	
11	Appendix I - Screenshot	
12	Appendix II - Github Profile	
13	Appendix III - AWS Course Completion certificate	

BONAFIDE

This is to certify that 18CSC209J - DATABASE MANAGEMENT SYSTEM AND CLOUD INTEGRATION SERVICES LABORATORY Mini Project report titled “ Book Bank Management” is the bonafide work of Sumith Sai K (RA2111028010164), Simran Pradhan (RA2111028010190), Naresh K (RA2111028010158) who undertook the task of completing the project within the allotted time.

SIGNATURE

SIGNATURE

Dr. Angayarkanni

Dr. Annapuran

Panaiyappan K

Assistant Professor
Department of Networking
and Communication
SRM Institute of Science
and Technology

Professor and Head
Department of Networking
and Communication
SRM Institute of Science
and Technology

Abstract:

This project titled Book Bank Management System is focused on designing and implementing Database Management System in which a member can register themselves and then he can borrow books from book bank. It mainly concentrates on providing books for engineering students.

The main objective of the database was to design a Online book bank system enable a central monitoring mechanism of the book-bank be more faster and less error prone. The database contains different types of tables like Authentication, Book, Registration, Subscription and User. It also follows sorting of fields and perform various functions using different commands.

The project will be implemented using sql plus that is used for creating tables and assigning values to different attributes and records in table. This database will be connected to the creation of tables in DynamoDB. Overall, this project aims to provide a reliable and efficient DBMS for Book Bank Management System that can handle a large volume of registration as well as subscription by users while maintaining data authentication and security.

Introduction:

This system would be used by members of any college to check the availability of the books and borrow the books, and then the databases are updated. This document aims to analyze and elaborate on the high-level needs and features of the book bank management system. It also tells the usability, and reliability defined in the use case specification.

Objective:

The system's main objective was to design an online book-bank monitoring system to enable a central monitoring mechanism of the book-bank to be faster and less error-prone. Apart from this,

- To help the students acquire the right books for the syllabus at the right time.
- To ensure availability of basic textbooks to students against limited funds.
- To develop students' ability to handle property loaned to them.

Over View :

The overview of this project is to design a tool for book bank so that it can be used by any book banks to lend their books as well as colleges.

Attributes used in the diagrams (GLOSSARY) :

1. **MEMBER** : The one who registers himself and purchase books from the bank.
2. **DATABASE**: Database is used to store the details of members and books.
3. **ADMINISTRATOR**: The one who verifies the availability of book and issue them
4. **USER** : Member
5. **SOFTWARE REQUIREMENT SPECIFICATION**:
This software specification documents full set of features and function for online recruitment system that is performed in company website.

Requirements :

- ❖ The BOOK BANK MANAGEMENT SYSTEM(BBMS) is accessed by only the registered users.
- ❖ The BBMS accepts the user id from the user.
- ❖ The BBMS shall accept the user selection for a range of books buying transactions.
- ❖ The BBMS is operated on search mechanism which is on behalf of the Name of the book, Category, Author, Issued date , Issued organisation.
- ❖ If the user want to buy books then the user get the receipt of the number of books and other info after the payment checkout.
- ❖ User can also have an membership account of BBMS which helps for easy and effective use of BBMS.
- ❖ The BBMS stores all the details about the books and all the transactions done by the BBMS users.

Identification of modules :

Sections Accounting System: This system contains the information regarding the books that each section of Book Bank contains.

Database: This contains all the book details, transactions , user details aswell as upcoming books publishing details and other information about the books.

Section Counter System : This System Contains the all the information of the books present in all various section of counters and manages the books count according to the books usage and tries to improve the user experience.

Maintenance System: This system handles all the requirements of the books and the book bank database for example adding the new user,book details along with updating the user details and the updation of new version of books with old versions and making all books available for the readers.

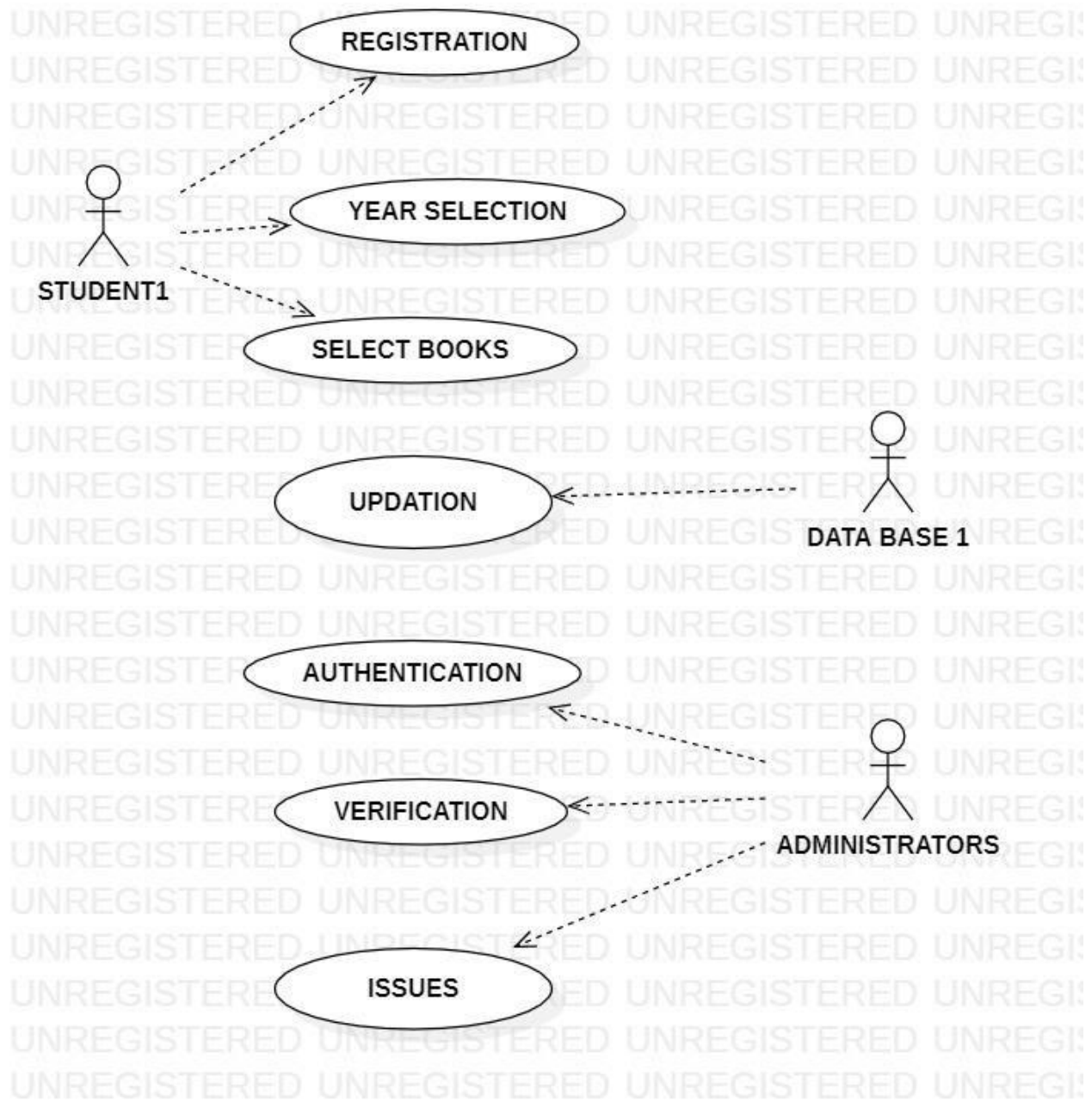
Security system: This system takes care of the books thefts and smooth transaction in the payments in a secured manner.

System Design

Use case diagram:

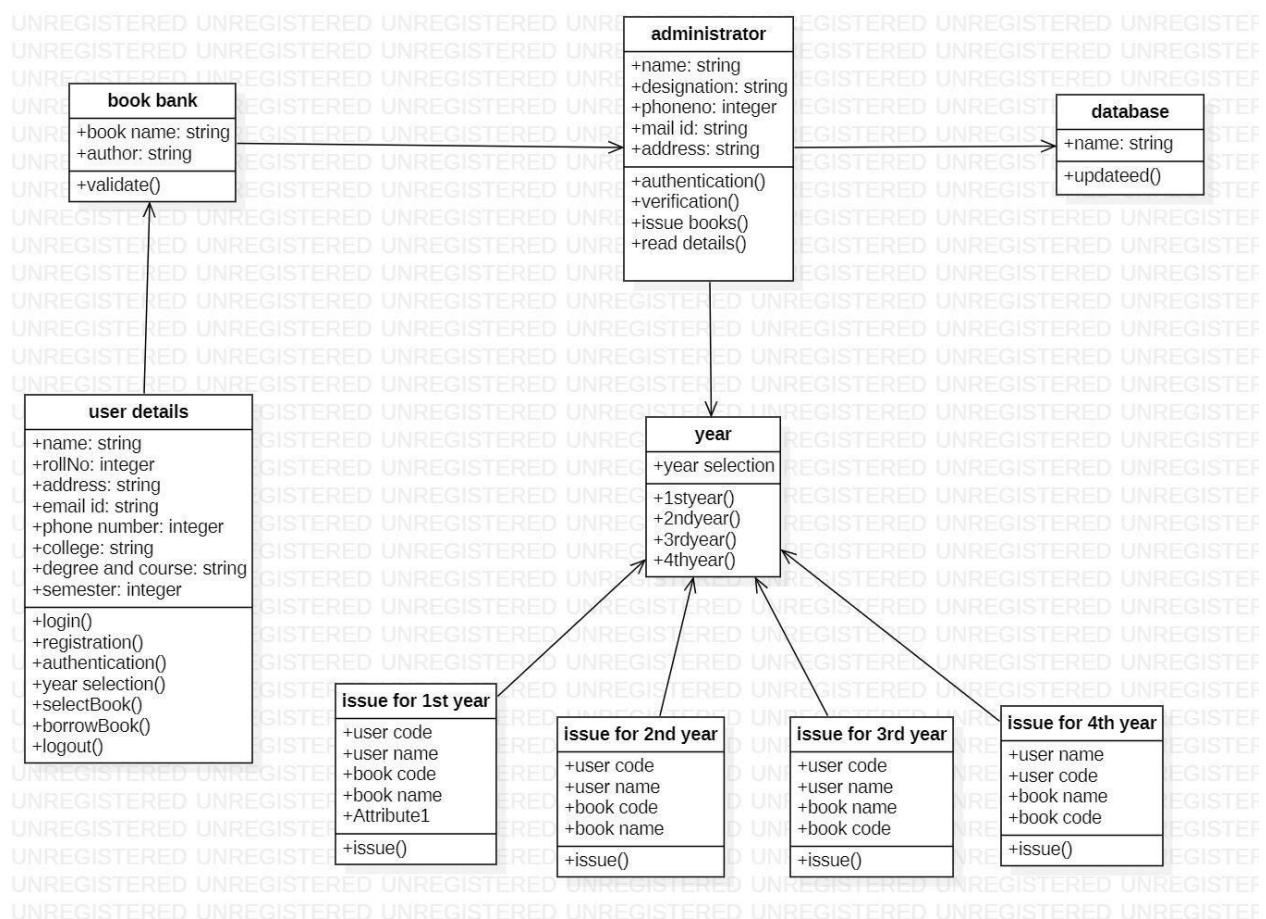
A use case diagram is a diagram which consists of set of use cases and actors enclosed by system boundary, and association between use cases and actors. Use cases diagram especially important in organizing, modelling the behaviour of the system.

Use case is a set of scenarios tied together by a common user goal. A scenario is a sequence of steps describing the interaction between a user and system.



Class Diagram

This Class Diagram describes that set of classes, interfaces, and collaboration and their relationship. It describes the types of objects in their system and various kinds of static relationship that exceeds among them.



ER DIAGRAM:

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database.

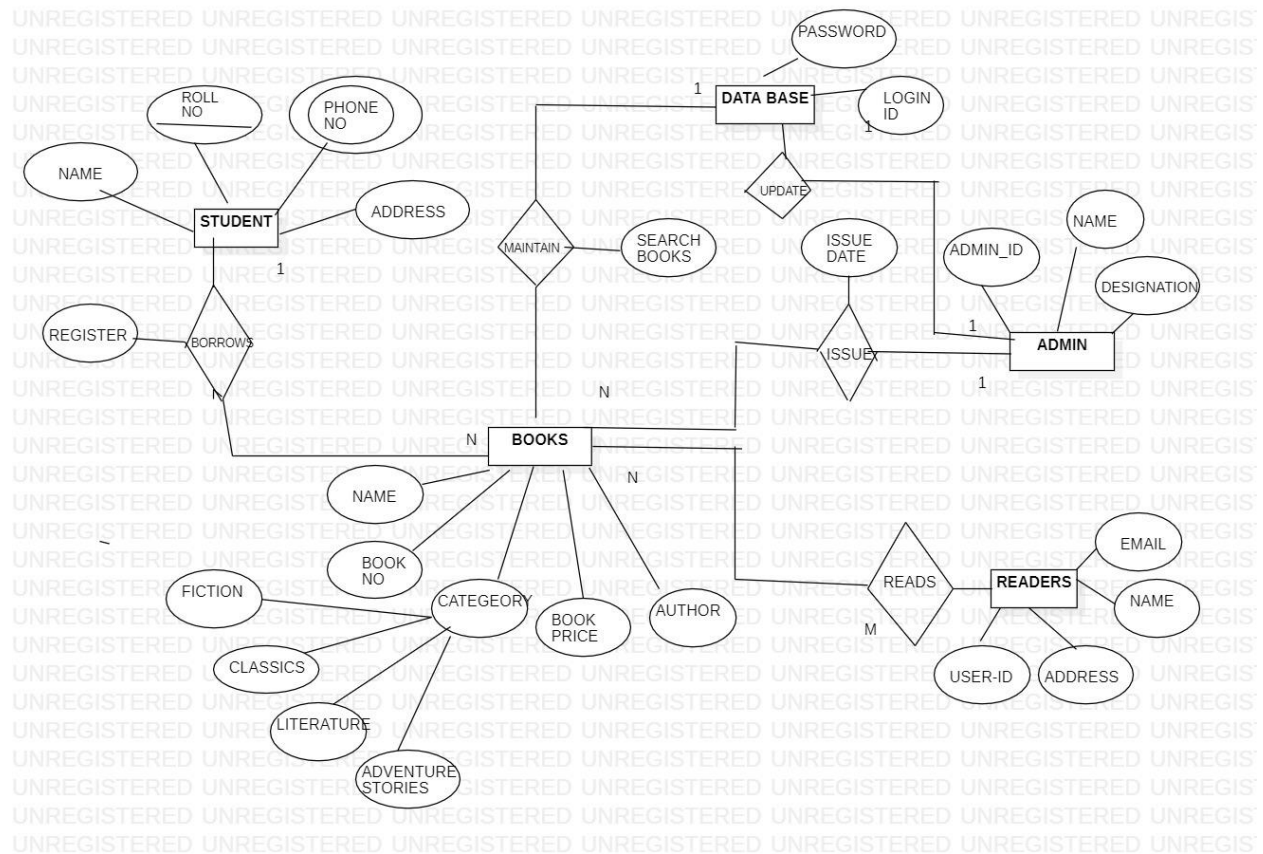
ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

ER model helps to systematically analyze data requirements to produce a well-designed database. The ER Model represents real-world entities and the relationships between them.

Creating an ER Model in DBMS is considered as a best practice before implementing your database.

ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes, and diamond shapes to represent relationships.

The purpose of ER Diagram is to represent the entity framework infrastructure.



Database Creation using DDL and DML:

We use the DDL commands for creating the database or schema, while DML commands are used to populate and manipulate the database. DDL commands can affect the whole database or table, whereas DML statements only affect single or multiple rows based on the condition specified in a query.

For Book Bank Management System by using DDL commands we can create the databases like User, Registration, Subscription, Books, Authentication etc and by using DML commands we can insert, update, delete ,commit and rollback the values in each table.

Creating table for User:

```
CREATE TABLE user (  
user_id INT NOT NULL PRIMARY KEY,  
Name VARCHAR(50),  
Address VARCHAR(50),  
Mobile_Number VARCHAR(20),  
email_address VARCHAR(50),  
Payment_Type VARCHAR(50)  
);
```

--Insert values into User table

```
INSERT INTO user (user_id, Name, Address, Mobile_Number, Payment_Type,  
email_address)  
VALUES  
(1, 'Sumith', 'Hyderabad', '9849345652', 'Debit Card','km0686@srmist.edu.in'),  
(2, 'Naresh Kothoju', 'Hyderabad', '9701999548', 'UPI','kk3159@srmist.edu.in'),  
(3, 'Simran Pradhan', 'Odisha', '9668148716', 'Cash','sp5006@srmist.edu.in');
```

User					
user_id	Name	Address	Mobile_Number	email_address	Payment_Type
1	Sumith	Hyderabad	9849345652	km0686@srmist.edu.in	Debit Card
2	Naresh Kothoju	Hyderabad	9701999548	kk3159@srmist.edu.in	UPI
3	Simran Pradhan	Odisha	9668148716	sp5006@srmist.edu.in	Cash

Creating table for Registration:

```
CREATE TABLE registration (  
registration_id INT NOT NULL PRIMARY KEY,  
Name VARCHAR(50),  
Address VARCHAR(50),  
Mobile_Number VARCHAR(20),  
email_address VARCHAR(50)  
);
```

--Insert values into Registration table

```
INSERT INTO registration (registration_id, Name, Address,
Mobile_Number,email_address)
VALUES
(1, 'Sumith', 'Hyderabad', '9849345652','km0686@srmist.edu.in'),
(2, 'Naresh Kothoju', 'Hyderabad', '9701999548','kk3159@srmist.edu.in'),
(3, 'Simran Pradhan', 'Odisha', '9668148716','sp5006@srmist.edu.in');
```

Registration				
registration_id	Name	Address	Mobile_Number	email_address
1	Sumith	Hyderabad	9849345652	km0686@srmist.edu.in
2	Naresh Kothoju	Hyderabad	9701999548	kk3159@srmist.edu.in
3	Simran Pradhan	Odisha	9668148716	sp5006@srmist.edu.in

Creating table for Subscription:

```
CREATE TABLE subscription (
Subscription_id INT NOT NULL PRIMARY KEY,
subscriber_name VARCHAR(50),
Date_of_Subscription DATE,
Date_of_Expiry DATE
);
```

```
--Insert values into Subscription table
INSERT INTO subscription (Subscription_id, subscriber_name,
Date_of_Subscription, Date_of_Expiry)
VALUES
(3, 'Simran Pradhan', '12-12-2022', '12-12-2024');
```

Subscription			
Subscription_id	subscriber_name	Date_of_Subscription	Date_of_Expiry
1	Simran Pradhan	12-12-2022	12-12-2024

Create Book table:

```
CREATE TABLE book (
```

```

Book_id INT NOT NULL PRIMARY KEY,
Book_Name VARCHAR(50),
Author VARCHAR(50),
Genre VARCHAR(50),
Language VARCHAR(50),
Year INT
);

```

-- Insert values into Book table

```

INSERT INTO book (Book_id, Book_Name, Author, Genre, Language, Year)

VALUES

```

```

(1, 'Cosmos', 'Carl Sagan', 'Science', 'English', '1980'),
(2, 'How The Mind Works', 'Steven Pinker', 'Non Fiction', 'English', '1997'),
(3, 'A Tale Of Two Cities', 'Charles Dickens', 'Novel, Historical Fiction, historical
novel', 'English', '1859'),
(4, 'The Hobbit', 'J. R. R. Tolkien', 'Novel, Fantasy Fiction, High fantasy, Childrens
literature, Epic, Epic Fiction', 'English', '1937'),
(5, 'She, a History of Adventure', 'H. Rider Haggard', 'Adventure fiction, Fantasy
Fiction, Romance novel, Gothic fiction, Romance, Chivalric romance',
'English', '1886');

```

Book					
Book_id	Book_Name	Author	Genre	Language	Year
1	Cosmos	Carl Sagan	Science	English	1980
2	How The Mind Works	Steven Pinker	Non Fiction	English	1997
3	A Tale Of Two Cities	Charles Dickens	Novel, Historical Fiction, historical novel	English	1859
4	The Hobbit	J. R. R. Tolkien	Novel, Fantasy Fiction, High fantasy, Childrens literature, Epic, Epic Fiction	English	1937
5	She, a History of Adventure	H. Rider Haggard	Adventure fiction, Fantasy Fiction, Romance novel, Gothic fiction, Romance, Chivalric romance	English	1886

Create Authentication Table:

```
CREATE TABLE authentication (  
  Username VARCHAR(50),  
  Password VARCHAR(50)  
);
```

-- Insert values into Authentication table

```
INSERT INTO authentication ( Username, Password)  
VALUES  
( 'Sumith', 'RA2111028010164'),  
( 'Naresh Kothoju', 'RA2111028010158'),  
( 'Simran Pradhan', 'RA2111028010190');
```

Authentication	
Username	Password
Sumith	RA2111028010164
Naresh Kothoju	RA2111028010158
Simran Pradhan	RA2111028010190

Normalization of Database:

Normalization is the process for evaluating and correcting table structures and determines the optimal assignments of attributes to entities as well as provides micro view of entities.

- Reduce uncontrolled data redundancies and help to eliminate data anomalies.
- Produce controlled redundancies to link tables.

There are several levels of normalization, with each level building upon the previous one. The most commonly used levels are:

- **First Normal Form (1NF):** This level requires that each table has a primary key and that all attributes are dependent on the primary key.

For a Book Bank management system, the user information can be stored in a single table with columns such as user_id, name, address, mobile number, payment type and subscriber. The user_id can serve as the primary key for this table.

- **Second Normal Form (2NF):** This level requires that all non-key attributes are dependent on the entire primary key and there are no partial dependencies.

In Book Bank Management System, the user information table may need to be split into two tables to satisfy 2NF. One table could contain the user_id, name, address, mobile number, while the other table could contain the payment type and subscriber. This ensures that user information is not duplicated and avoid redundancy.

- **Third Normal Form (3NF):** This level requires that all non-key attributes are not dependent on any other non-key attributes and there are no transitive dependencies.

For a Book Bank Management System, the subscription table may need to split into two tables to satisfy 3NF. One table could contain Subscriber's name , date and expiry of subscription , while the other table could contain the subscriber validity. This ensures that customer information is not duplicated and that changes to one record do not affect other records

By following the normalization process, a Book Bank Management system can be designed to efficiently store user, book and subscription information, making it a reliable and secure book borrow service for users.

Implementation using Dynamo DB:

DynamoDB is a NoSQL database service offered by AWS that is designed to provide fast and scalable performance. For a Book Bank Management System , DynamoDB can be used to store user information, Database of Books, Subscription details, Authentication and Registration data.

1. Define the data model: Determine the data you need to store for a Book Bank Management System. For example, you may need to store information about User details like user id, User Info and Book Transaction details.

2. Create a DynamoDB table: Create a table in DynamoDB to store the data. Define the primary key for the table, which could be a single partition key or a combination of a partition key and a sort key.

3. Define the table schema: Define the attributes that you want to store in the table. You can also specify any secondary indexes that you want to create.

Here is a possible implementation for a Book Bank Management System using DynamoDB:

Create a table for user information:

The table can have the following attributes:

User ID (partition key)

Name (Sort Key)

Address

Mobile Number

Payment Type

Email address

Create a table for registration:

The table can have the following attributes:

Registration ID(partition Key)

Name (Sort Key)

Address

Mobile Number

Payment Type

Email address

Create an IAM role with appropriate permissions for accessing DynamoDB.

For user information: you can use the DynamoDB API to create, read, update and delete records in the table. You can also use the Query operation to retrieve user account information by user id.

For transaction history: you can use the Put Item operation to add a new transaction to the table, and the Query operation to retrieve transactions for a specific user id within a given time range.

To ensure data consistency and prevent race conditions, use conditional writes to update account information. For example, when processing a query, you can use a conditional write to ensure that the subscription is there for a specific user or not. Use DynamoDB streams to capture changes to the transaction history table, and trigger appropriate actions such as sending subscription alerts or updating subscription before the expiry of it.

Use DynamoDB's scalability features, such as auto-scaling and partitioning, to ensure that the system can handle increasing traffic and transaction volumes. In summary, DynamoDB can be a reliable and scalable choice for implementing an Book Bank Management system, with the flexibility to store and retrieve data quickly and efficiently, and handle high levels of traffic and transaction volumes.

Conclusion:

The Book Bank Management System project in DBMS is a comprehensive system that aims to provide secure and convenient book bank services to users. The system is designed with a robust database management system that ensures the security, integrity, and consistency of data. Overall, the Book Bank Management System project in DBMS is an excellent solution for users who are curious to avail all the types of genres and categories of books within less time efficiently and effectively through the Book Bank Management System.

Appendix I :

The screenshot shows the AWS Management Console interface for the Mumbai region. The top navigation bar includes the AWS logo, Services menu, a search bar containing "[Alt+S]", and user information for SUMITH SAI KORABINA. The left-hand navigation pane lists various services like Dashboard, IAM, Lambda, etc., with "DynamoDB" selected. Under the "DynamoDB" section, "Tables" is chosen. The main content area displays the "Tables (5) Info" view. It features a search bar with the placeholder "Find tables by table name" and a dropdown for "Any table tag". Below this is a table listing five existing tables: User, Subscription, Registration, Book, and Authentication. Each row provides details such as its status (Active), partition key, sort key, indexes, deletion protection, and capacity mode.

	Name	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode	Write capacity mode
<input type="checkbox"/>	User	Active	user_id (S)	Name (S)	0	Off	Provisioned with auto scaling (1)	Provisioned with auto scaling (1)
<input type="checkbox"/>	Subscription	Active	Subscription_Id (S)	Subscriber_Name (S)	0	Off	Provisioned with auto scaling (1)	Provisioned with auto scaling (1)
<input type="checkbox"/>	Registration	Active	Registration_ID (S)	Name (S)	0	Off	Provisioned with auto scaling (1)	Provisioned with auto scaling (1)
<input type="checkbox"/>	Book	Active	Book_id (S)	Book_name (S)	0	Off	Provisioned with auto scaling (1)	Provisioned with auto scaling (1)
<input type="checkbox"/>	Authentication	Active	username (S)	-	0	Off	Provisioned with auto scaling (1)	Provisioned with auto scaling (1)

Services

Search

[Alt+S]

Mumbai

SUMITH SAI KORABOINA

DynamoDB

Tables

Update settings

Explore items

PartiQL editor

Backups

Exports to S3

Imports from S3

Reserved capacity

Settings

DAX

Clusters

Subnet groups

Parameter groups

Events

DynamoDB > Tables > User

Tables (5)

Any table tag

Find tables by table name

1

User

Subscription

Registration

Book

Authentication

User

Overview

Indexes

Monitor

Global tables

Backups

Exports and streams

Additional settings

Protect your DynamoDB table from accidental writes and deletes

When you turn on point-in-time recovery (PITR), DynamoDB backs up your table data automatically so that you can restore to any given second in the preceding 35 days. Additional charges apply. [Learn more](#)

Edit PITR

General information

Partition key

user_id (String)

Sort key

Name (String)

Capacity mode

Provisioned

Table status

Active

Alarms

No active alarms

Point-in-time recovery (PITR)

Off

Additional info

Items summary

DynamoDB updates the following information approximately every six hours.

Get live item count

Item count

3

Table size

258 bytes

Average item size

86 bytes

Services

Search

[Alt+S]

Mumbai

SUMITH SAI KORABOINA

DynamoDB

Tables

Update settings

Explore items

PartiQL editor

Backups

Exports to S3

Imports from S3

Reserved capacity

Settings

DAX

Clusters

Subnet groups

Parameter groups

Events

DynamoDB > Tables > User

Tables (5)

Any table tag

Find tables by table name

1

User

Subscription

Registration

Book

Authentication

Scan or query items

Scan

Query

Select a table or index

Table - User

Select attribute projection

All attributes

user_id (Partition key)

3

Name (Sort key)

Equal to

Simran Pradhan

Sort descending

Filters

Run

Reset

Completed. Read capacity units consumed: 0.5

Items returned (1)

user_id

Name

Address

Mobile Number

Payment Type

Subscriber

3

Simran Pradhan

Odisha

9668148716

UPI

true

Appendix II :

Sumith - <https://github.com/SUMITHSAI>

The screenshot shows the GitHub profile page for user SUMITHSAI. The profile includes a circular avatar with a pink and white pixelated design, the username SUMITHSAI, and an 'Edit profile' button. Below the profile information is a 'Highlights' section with a 'PRO' badge. The main content area displays a list of repositories under the 'Repositories' tab, which is currently showing 8 items. The repositories listed are:

- PL-SQL-Conditional-and-Iterative-statements** (Private): Updated 4 minutes ago.
- PL-SQL-Procedures** (Private): Updated 5 minutes ago.
- PL-SQL-Functions** (Private): Updated 6 minutes ago.
- PL-SQL-Cursors** (Private): Updated 9 minutes ago.
- Calculator** (Public): Updated 2 weeks ago.
- Bus-Ticket-Booking** (Public): Updated on Jun 30, 2022.
- CT1-MiniProject** (Public): Updated on Apr 26, 2022.
- SUMITHSAI** (Public): Config files for my GitHub profile. Updated on Apr 11, 2022.

Each repository entry includes a star button and a dropdown menu. The footer of the page contains links for Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About, along with the GitHub logo and copyright notice: © 2023 GitHub, Inc.

APPENDIX III

AWS Course Completion Certificate



AWS Course Completion Certificate



AWS Course Completion Certificate

