



DBMS Project :

LIBLIVE●

library management online on phone

2 0 2 4

A project by
Manik Chauhan [RA2211003030245]
Arya Gupta [RA2211003030247]
Khushi Joshi [RA2211003030249]
Vishwas Upadhyay [RA2211003030255]

LibLive

Introduction:

1

2

3

4

5

6

The essence of a library lies in its ability to disseminate knowledge effectively. However, the digital tide has altered the landscape of information exchange, rendering traditional library management systems inadequate. In response to this challenge, we embarked on a project to design an innovative Library Management System (LMS) that integrates the relational database capabilities of MySQL with the powerful scripting of Python, interfacing with users through a WhatsApp Twilio chatbot.



1

2

3

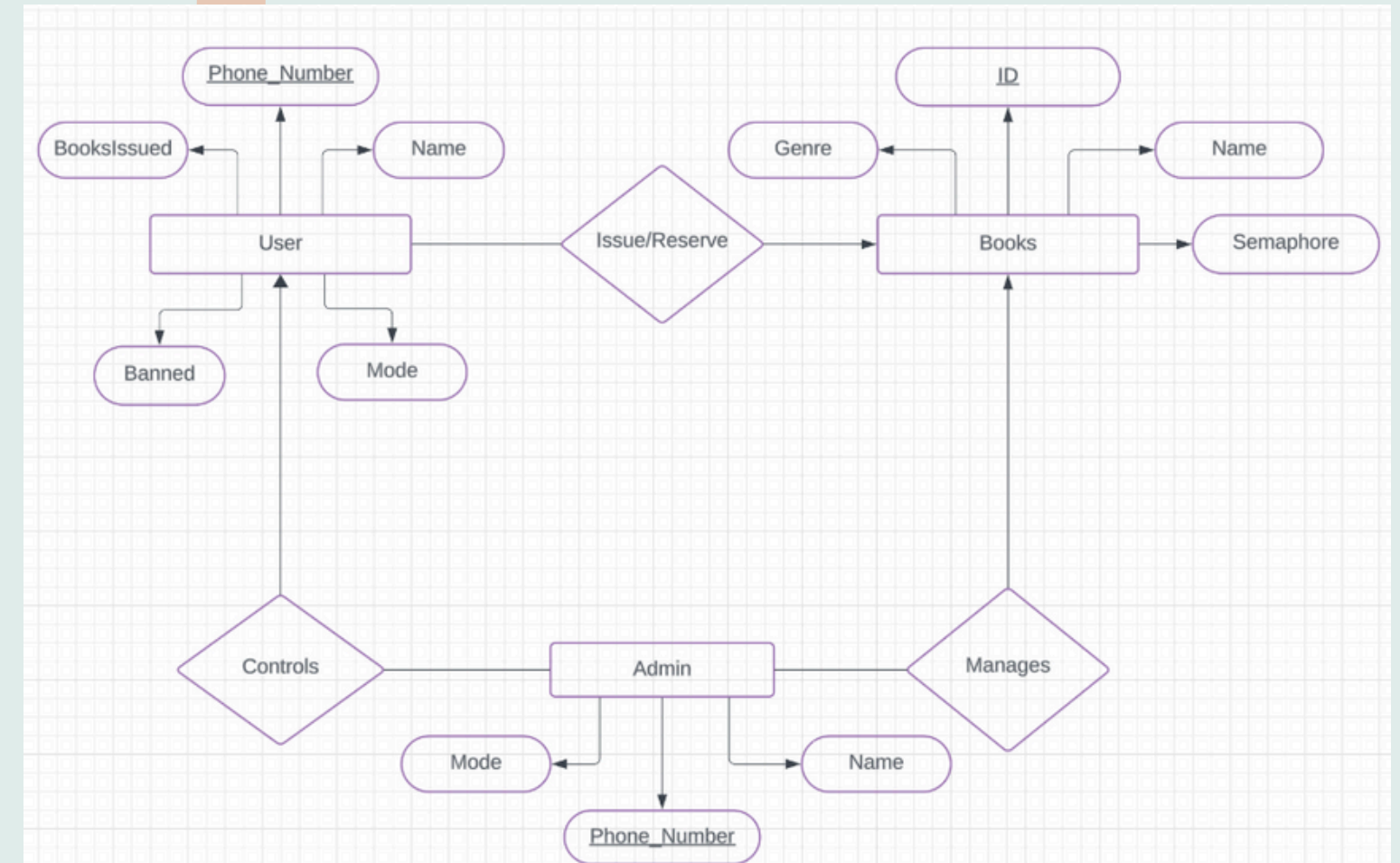
4

5

6

ER DIAGRAM

Here we designed an Entity-Relationship (ER) diagram for a library management system. We began by identifying the primary entities that are integral to the library system: User, Books, Admin, and the relationship entities Issue/Reserve, Controls, and Manages. These entities were analyzed for their attributes and relationships with one another.



1

2

3

4

5

6

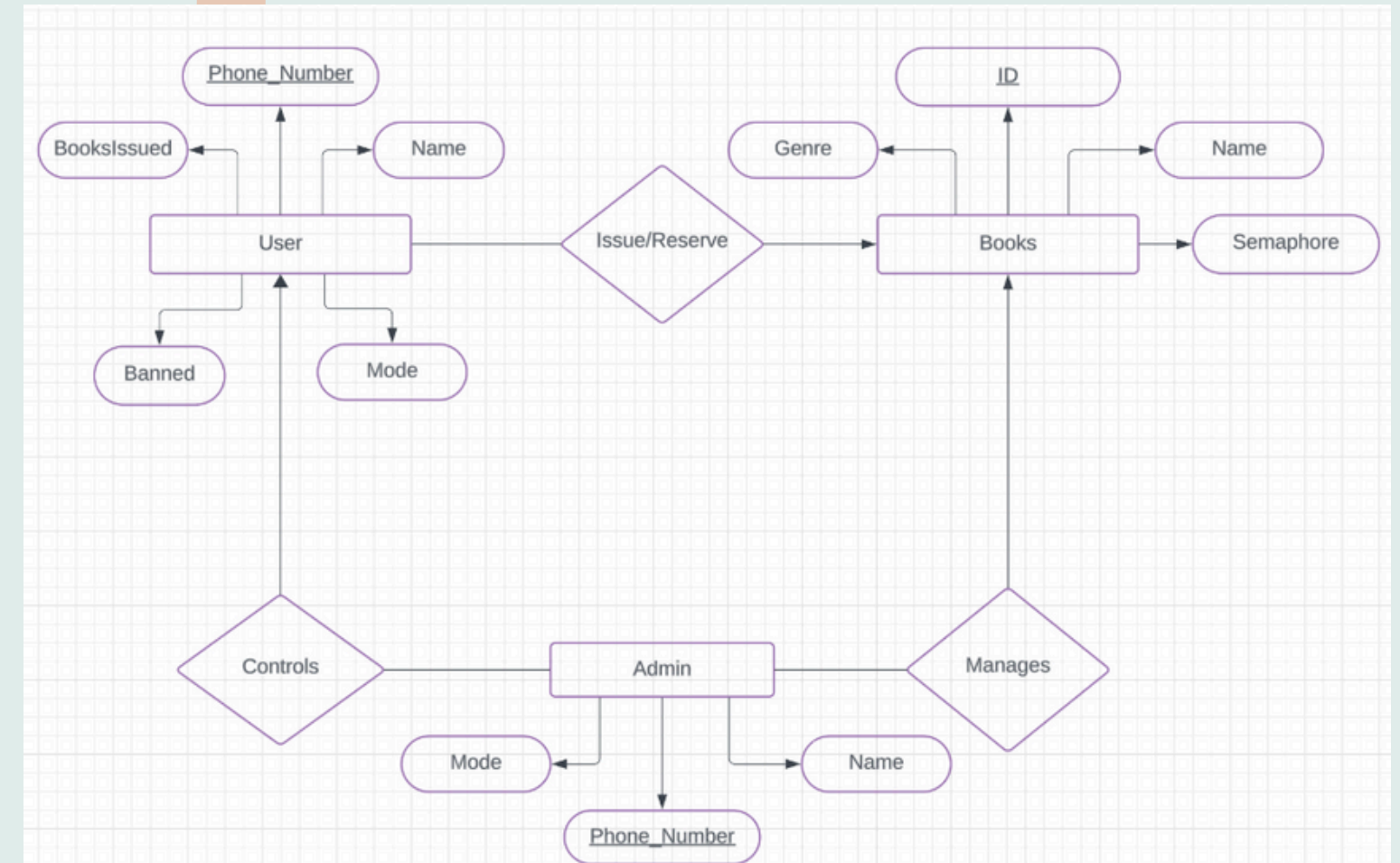
ER DIAGRAM

Entities and Attributes

User: This entity represents the library's patrons and includes attributes such as Name, Phone_Number, BooksIssued, Banned, and Mode. These attributes allow the system to store personal details, contact information, and the current process status of the user in the library system.

Books: This entity is critical for cataloging purposes and includes ID, Name, Genre, and Semaphore. These attributes help in identifying books, categorizing them by genre, and maintaining a semaphore for tracking book availability.

Admin: The admin entity is designed to handle administrative tasks and has attributes like Name, Phone_Number, and Mode. It facilitates contact with library staff and denotes their operational mode within the system.



1

2

3

4

5

6

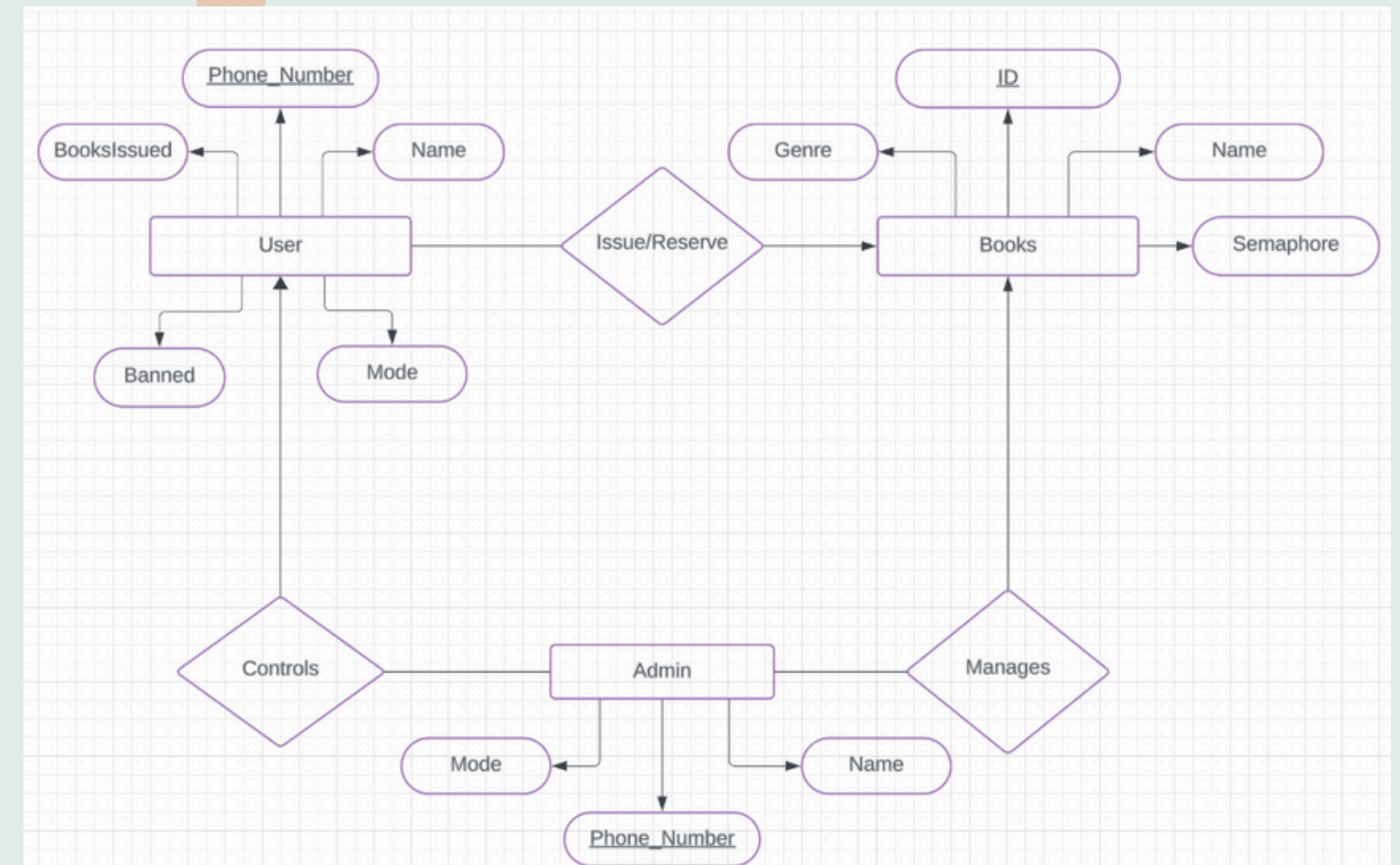
ER DIAGRAM

Relationships

Issue/Reserve: This relationship entity links User and Books to depict the action of issuing or reserving books. It is a dynamic entity that reflects the current transactions between users and the library's inventory

Controls: This relationship entity illustrates the control that Admin has over User actions, influencing whether a user is banned or allowed certain access within the library system.

Manages: Another relationship entity that connects Admin to Books, signifying the administrative tasks of managing the book inventory, such as adding new books, updating book status, and catalog maintenance.



Conversion of ER diagram to Relational Model

1

2

3

4

5

6

The process began with the deconstruction of the ER diagram into its constituent entities and associations. Each entity was then translated into a relational table, with entity attributes becoming table fields. Data types were carefully assigned to each field to ensure integrity and efficiency of data storage. The One to Many Relationship between entities were converted by adding the attributes of the relation to the 'Many' Relationship Entity with a Foreign key connecting to the 'One' Relationship Entity.

Admin:

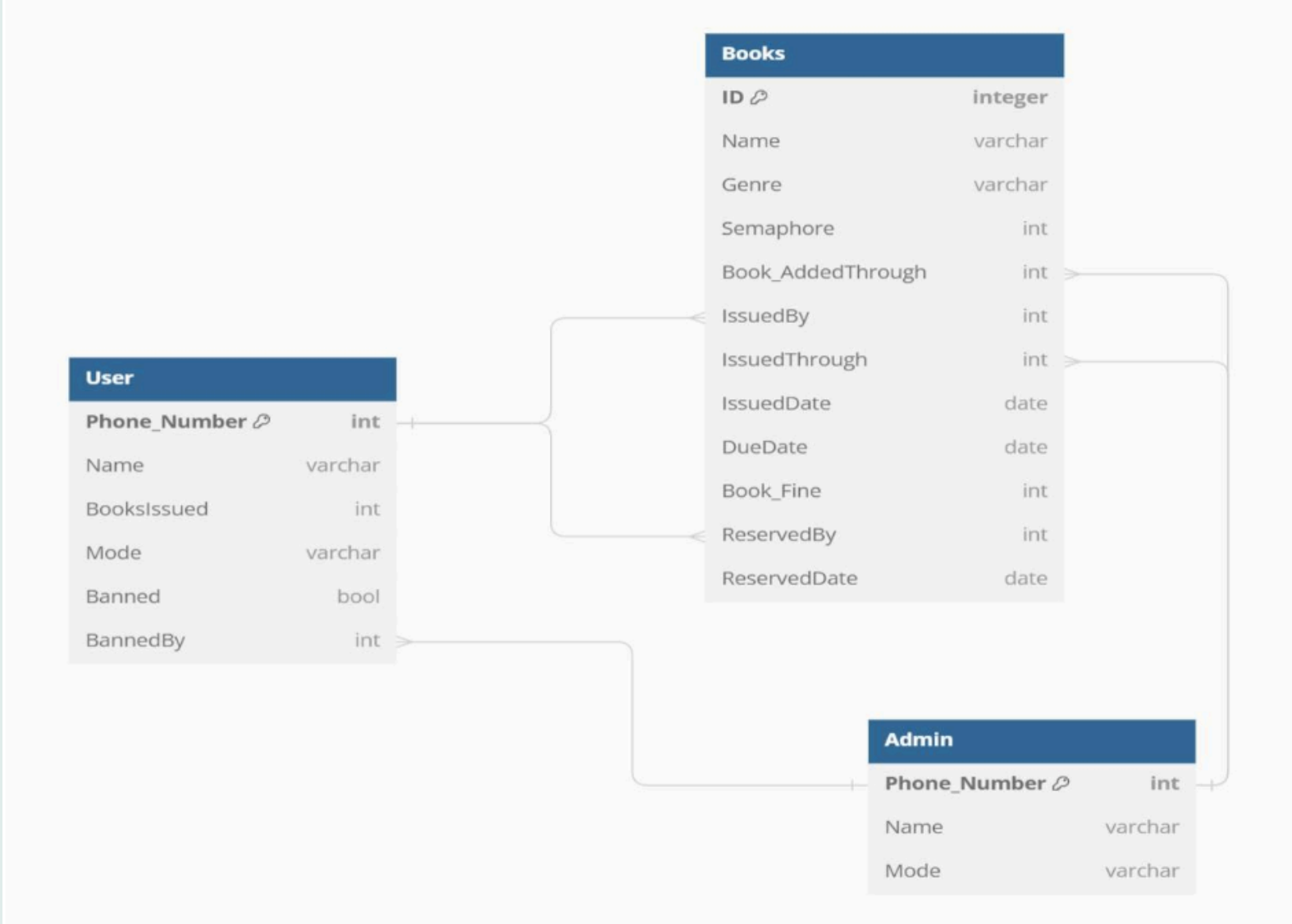
Phone_Number	Name	Mode
+919560405533	Harshit	IssueMenu
+919999367202	Vibhor	AdminMenu
NULL	NULL	NULL

Books:

ID	Name	Genre	Semaphore	Book_AddedThrough	IssuedBy	IssuedThrough	IssuedDate	DueDate	Book_Fine	ReservedBy	ReservedDate
8	Spiderman	Action	1	+919999367202	NULL	NULL	NULL	NULL	0	NULL	NULL
9	Dragon Ball	Action	1	+919999367202	NULL	NULL	NULL	NULL	0	NULL	NULL
10	Demon Slayer	Action	1	+919999367202	+919667040342	+9199993672...	2024-04-10	2024-05-06	0	NULL	NULL
11	Steins Gate	Action	1	+919999367202	NULL	NULL	NULL	NULL	0	NULL	NULL

User:

Phone_Number	Name	BooksIssued	Mode	Banned	BannedBy
+917882345621	Rahul	NULL	UserMenu	NULL	NULL
+919667040342	Rohit	NULL	UserMenu	1	+919999367202
NULL	NULL	NULL	NULL	NULL	NULL



1

2

3

4

5

6

Normalization

The original table structure contains multiple fields related to users, books, their issuance, reservation, and admin activities, all within a single table. This conglomeration of data leads to numerous issues such as redundancy, update anomalies, and a lack of scalability.

First Normal Form (1NF):

To achieve 1NF, we ensure that the table has a primary key and that all attributes contain atomic (indivisible) values. We might begin by separating book-related information from user-related details, resulting in individual tables where each row is unique, and repeating groups (such as multiple genres for a single book) are eliminated. This would involve creating tables like BookGenres, which relate books to their genres.

Second Normal Form (2NF):

For a table to be in 2NF, it must first satisfy 1NF and then move all fields that are not dependent on the primary key to separate tables. This would involve creating tables like Reserve and Issue, which stores information about which user has issued or reserved which book. The relationship between books and users would be based on foreign keys that reference the primary keys in the books and users tables.

Normalization Outcome

BookGenres Table: Created to separate book genres from the main books table to avoid repeating the book name for each genre. Each book can have multiple genres, and each genre can include many books, establishing a many-to-many relationship between books and genres.

ID	Genre
16	Action
16	Superhero

Issue Table: Extracted from the main books table to track the issuance of books independently. This table includes the user who has issued the book, the admin through whom the book was issued, and relevant dates (issued and due). It helps manage the circulation of books effectively without storing redundant user or book data in the books table.

Issue_ID	ID	IssuedBy	IssuedThrough	IssuedDate	DueDate	Book_Fine
6	16	+919871405533	+919999367202	2024-03-24	2024-04-09	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Reserve Table: Separated to manage reservations of books by users. It holds information about which user has reserved which book and the reservation date, ensuring that a book's availability is dynamically managed without data duplication.

Reservation_ID	ID	ReservedBy	ReservedDate
1	15	+919560405533	2024-04-25
NULL	NULL	NULL	NULL

1

2

3

4

5

6

Normalization

Old Books Table

ID	Name	Genre	Semaphore	Book_AddedThrough	IssuedBy	IssuedThrough	IssuedDate	DueDate	Book_Fine	ReservedBy	ReservedDate
8	Spiderman	Action	1	+919999367202	NULL	NULL	NULL	NULL	0	NULL	NULL
9	Dragon Ball	Action	1	+919999367202	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10	Demon Slayer	Action	1	+919999367202	+919667040342	+9199993672...	2024-04-10	2024-05-06	0	NULL	NULL
11	Steins Gate	Action	1	+919999367202	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



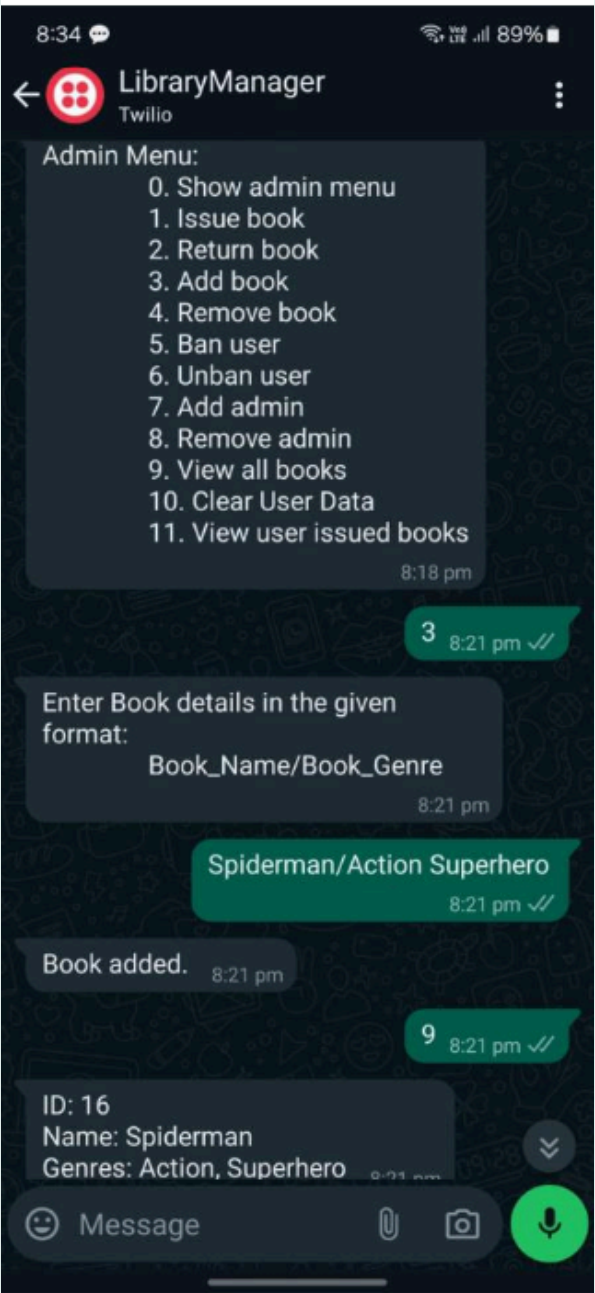
New Books Table

ID	Name	Semaphore	Book_AddedThrough
16	Spiderman	0	+919999367202
NULL	NULL	NULL	NULL

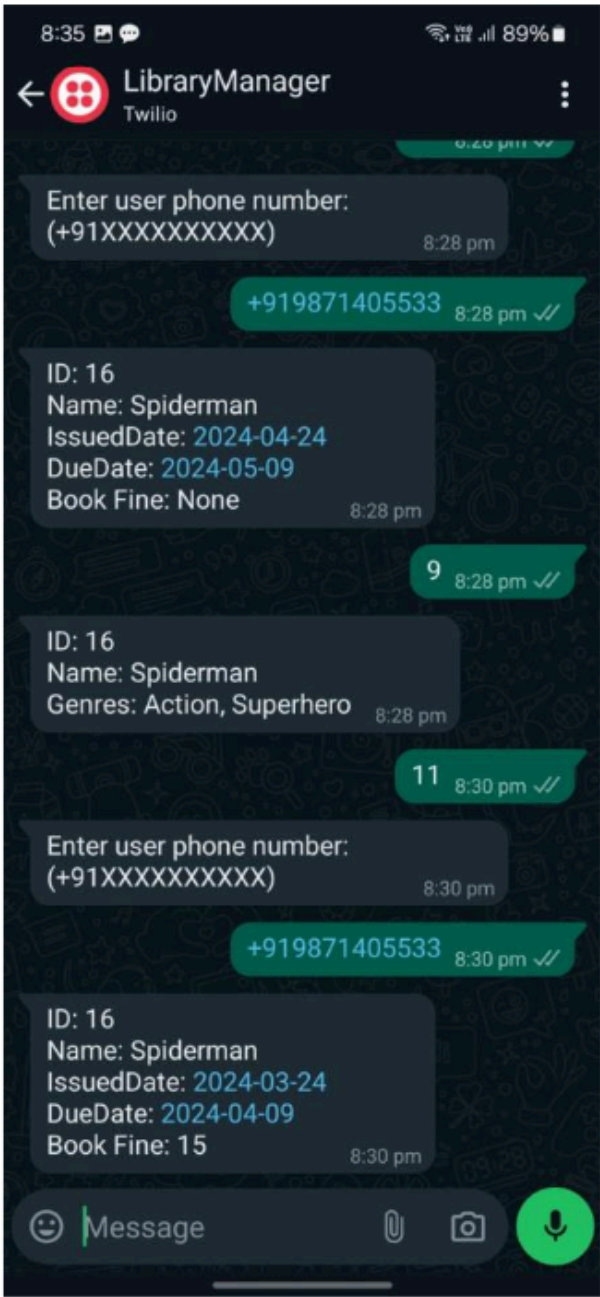
Results:

Admin

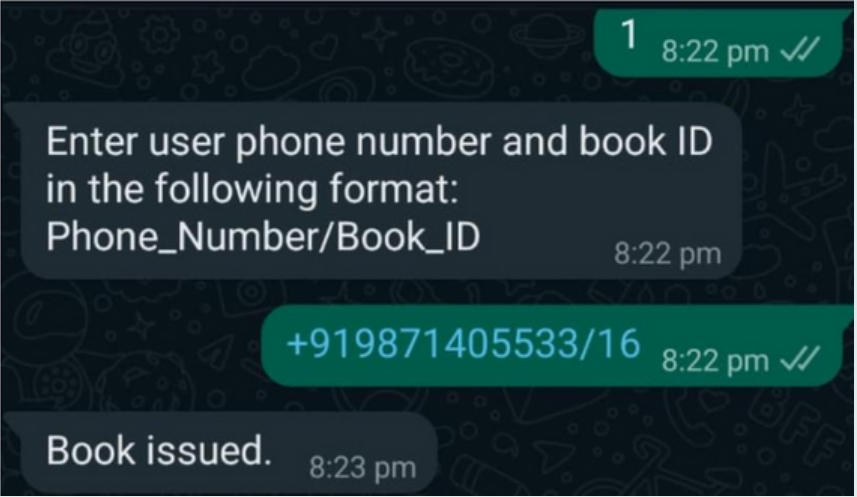
Adding Book:



Viewing issued book of user:



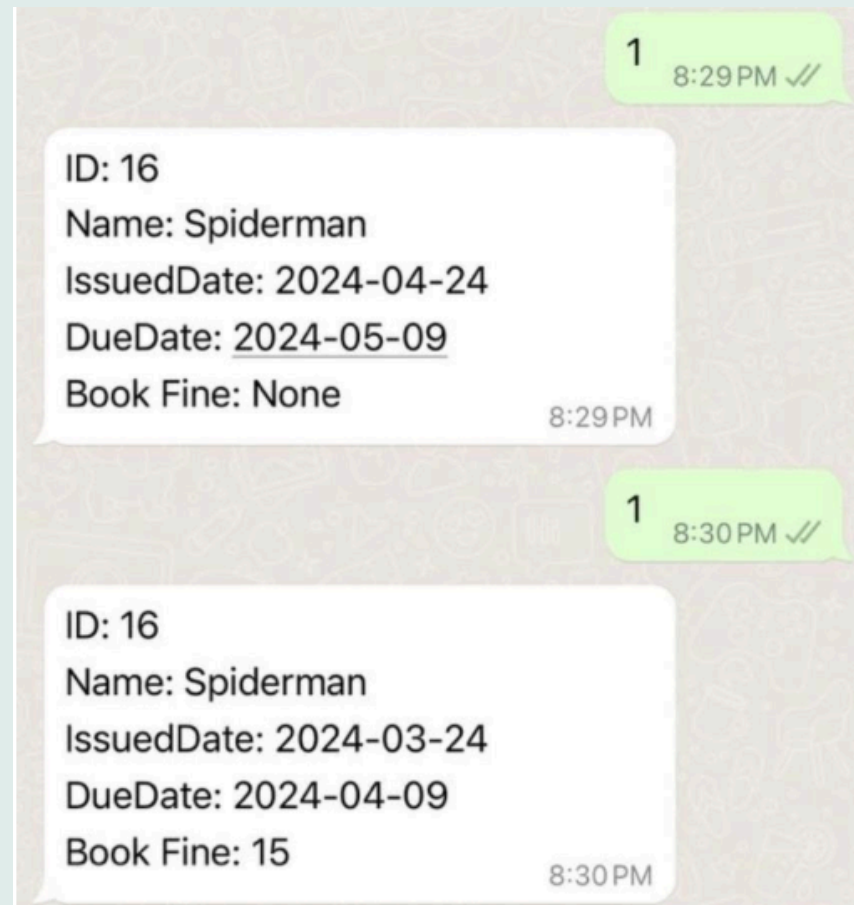
Issuing Book:



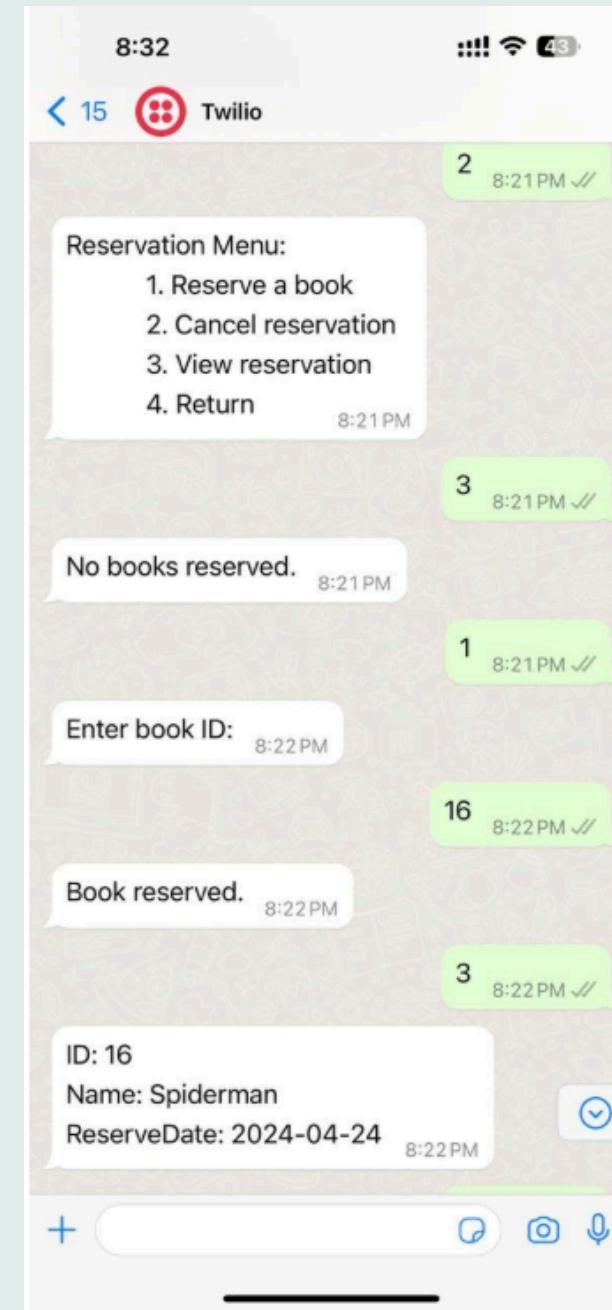
Results:

User

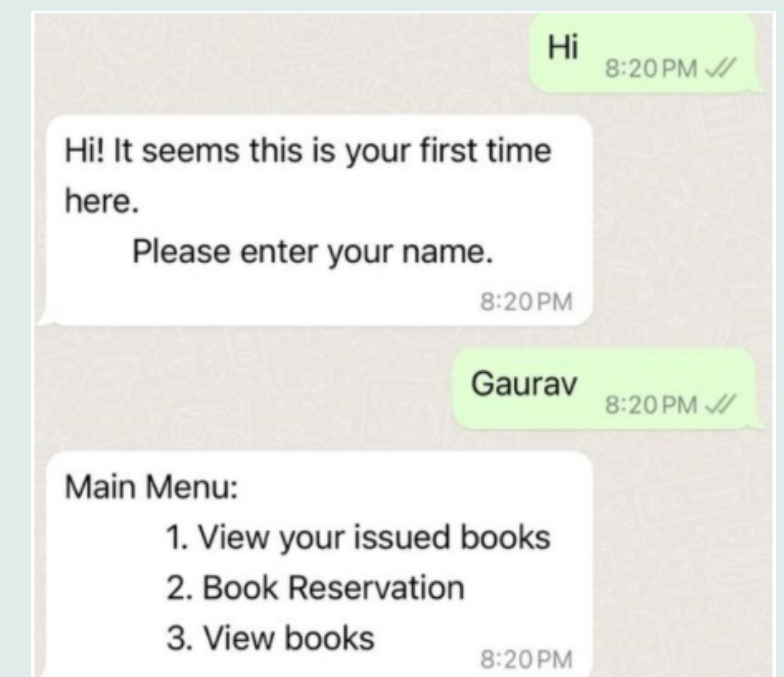
Viewing issued book after admin issues:



Reserve Menu and Reserving/Viewing reserved books:



First time User:



1

2

3

4

5

6

1

2

3

4

5

6



THANK YOU

