Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology
(Deemed to be University Estd. u/s 3 of UGC Act, 1956)
School of Computing
B.Tech. – Computer Science and Engineering

VTR UGE2021- (CBCS)

Academic Year: 2025–2026

SDG 4: Quality Education
Course Code : 10211CS207

Course Name : Database Management Systems

Slot No : S7-L1

DBMS USE CASE IMPLEMENTATION

Title: **A GIFT COUPON APPLICATION HANDLES OFFERS PAYMENT-RELATED INFORMATION.**

Submitted by:

VTUNO	REGISTER NUMBER	STUDENT NAME
VTU27850	24UECS1339	P.KOUSIK
VTU29302	24UECS1238	K.NITHISH KUMAR
VTU30108	24UECS0957	T.SOMU
VTU30103	24UECS1501	V.PUNEETH
VTU30270	24UECS1492	S.VENU
VTU30206	24UECS0099	G.SADA SIVA
VTU30213	24UECS0169	K.RAMA GOVARDHAN
VTU30196	24UECS1422	V.VISHNU VARDHAN

Under the guidance of:
Dr. T.RAVI,Associate Professor

INDEX **PAGE**

1. Introduction

In the rapidly evolving digital marketplace, businesses are constantly seeking innovative ways to attract, engage, and retain customers. Among the most effective methods is the use of gift coupons and promotional offers, which provide customers with exclusive discounts, rewards, and incentives. However, as the number of users and transactions increases, managing coupon issuance, validation, and redemption becomes a challenging task that requires a reliable, consistent, and scalable database management system.

Traditional coupon management systems often suffer from issues such as duplicate redemptions, inconsistent data, and slow transaction handling, especially when integrated with payment systems. These challenges can lead to financial losses, poor customer experience, and operational inefficiencies. To overcome these issues, the Gift Coupon Application introduces a relational database solution that ensures strong ACID compliance (Atomicity, Consistency, Isolation, and Durability) while supporting horizontal scalability for large-scale operations.

The primary goal of this application is to create a secure and efficient platform for businesses to manage coupons digitally. The system allows users to receive and redeem gift coupons seamlessly during purchases, ensuring that every transaction is processed accurately and securely. Each operation—from coupon creation and distribution to redemption and payment validation—is executed as a single atomic transaction, ensuring no duplication or inconsistency across the database.

To achieve this, the application employs a distributed relational database system such as CockroachDB or PostgreSQL with clustering capabilities. These databases provide strong consistency and fault tolerance, allowing the system to handle a high number of concurrent reads and writes without compromising transactional accuracy. By maintaining the ACID properties, the system guarantees that all coupon-related data, including user information, offer details, and payment records, remain synchronized and valid at all times.

In addition to relational data handling, the Gift Coupon Application incorporates MongoDB, a NoSQL database, for managing unstructured or semi-structured data such as promotional banners, campaign analytics, and user feedback. This hybrid integration ensures that while transactional data remains strongly consistent, analytical and dynamic content can scale freely, enhancing overall performance and flexibility.

Another significant aspect of this project is data normalization. The database schema is carefully designed and normalized up to Third Normal Form (3NF) to eliminate redundancy, prevent anomalies, and optimize query performance. By maintaining a normalized schema, the system can efficiently manage relationships between entities such as Users, Coupons, Offers, Payments, and Redemptions.

In essence, the Gift Coupon Application offers a modern, scalable, and secure database-driven solution that simplifies coupon lifecycle management while maintaining transactional reliability. It demonstrates how relational database principles, when combined with distributed architecture and selective NoSQL integration, can provide a powerful foundation for real-world e-commerce applications. This ensures that both customers and businesses benefit from a fast, accurate, and transparent coupon redemption experience.

2. Problem Statement

In today's fast-paced e-commerce environment, businesses rely heavily on digital coupons and promotional offers to attract customers. However, many existing coupon management systems struggle with data consistency, transaction reliability, and scalability. When multiple users attempt to redeem coupons simultaneously, issues such as duplicate redemption, inconsistent payment records, and transaction failures often arise.

Traditional centralized databases can become performance bottlenecks, while distributed NoSQL systems, though scalable, compromise strong consistency for speed through eventual consistency. Such trade-offs are unacceptable in applications that handle payment and financial data, where accuracy and ACID compliance are critical.

The Gift Coupon Application addresses these challenges by using a distributed relational database system that supports horizontal scaling without sacrificing ACID properties. This ensures every operation—coupon generation, redemption, and payment—is processed atomically and consistently, even under high concurrency.

To minimize redundancy and maintain integrity, the database schema is normalized up to Third Normal Form (3NF), establishing clear relationships between users, coupons, offers, payments, and redemptions. Additionally, MongoDB is integrated to manage non-transactional data such as promotional analytics and user feedback, providing flexibility and scalability for real-time insights.

In essence, the Gift Coupon Application aims to deliver a secure, consistent, and efficient database solution capable of handling large-scale coupon transactions with reliability and performance.

3. Objectives

• To design a secure and scalable relational database for managing user, coupon, and transaction data.

• To ensure ACID compliance for all financial and redemption operations.

• To provide horizontal scalability using a distributed relational database cluster.

• To maintain data accuracy and reduce redundancy through normalization up to 3NF.

• To demonstrate SQL and NoSQL integration, with MongoDB handling non-transactional data such as promotional banners or analytics..

4. System Requirements

Hardware Requirements:

- Processor: Intel i5 or higher
- RAM: 8 GB or more
- Hard Disk: 250 GB

Software Requirements:

- OS: Windows 10 or Linux
- Database: Oracle 12c or above
- Front-End: Java / Web Interface
- NoSQL: MongoDB 6.0
- Tools: Oracle SQL Developer, MongoDB Compass

5. System Analysis and Design

The system identifies the main entities and their relationships. Major entities include:
Main Entities Identified:

a. User

b. Coupon

c. Offer

d. Payment

e. Redemption

Relationships:

Each User can have multiple Coupons.

Each Coupon is linked to a specific Offer.

Each Redemption is associated with one Coupon and one Payment.

A Payment may be made by a User using one or more Coupons.

6. ER Diagram (Conceptual Design)

Relationships:

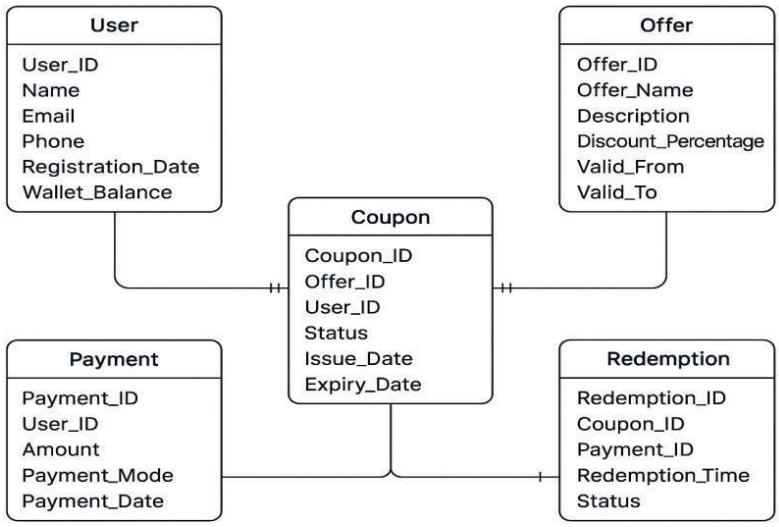
· User ↔ Coupon → One-to-Many

· Coupon ↔ Offer → Many-to-One

· Coupon ↔ Redemption → One-to-One

· Redemption ↔ Payment → Many-to-One

Figure: ER Diagram

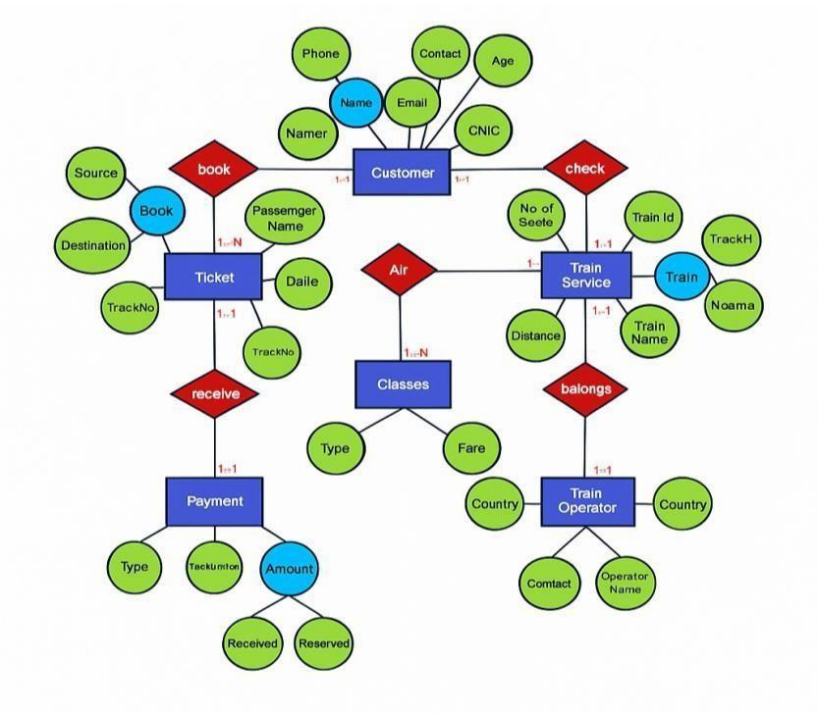


Entities and Attributes

1. User(User_ID, Name, Email, Phone, Registration_Date, Wallet_Balance)
2. Offer(Offer_ID, Offer_Name, Description, Discount_Percentage, Valid_From, Valid_To)
3. Coupon(Coupon_ID, Offer_ID(FK), User_ID(FK), Status, Issue_Date, Expiry_Date)
4. Payment(Payment_ID, User_ID(FK), Amount, Payment_Mode, Payment_Date)
5. Redemption(Redemption_ID, Coupon_ID(FK), Payment_ID(FK), Redemption_Time, Status)

Relationships:

- User ↔ Coupon → One-to-Many
- Coupon ↔ Offer → Many-to-One
- Coupon ↔ Redemption → One-to-One
- Redemption ↔ Payment → Many-to-One



}

---

4. Finding Redeemed Coupons

```
db.redemptions.find(
  { status: "Successful" },
  { coupon_id: 1, redeemed_on: 1, _id: 0 }
)
```

Output:

```
[
  { "coupon_id": "CPN001", "redeemed_on": "2025-10-20" },
  { "coupon_id": "CPN003", "redeemed_on": "2025-10-25" }]
```

12. Results and Discussion

The proposed Gift Coupon Application successfully manages coupon creation, validation, redemption, and payment processing using a normalized relational database. Each transaction maintains ACID properties, ensuring no duplicate or inconsistent redemptions occur.

The horizontal scalability of the relational cluster supports a large number of reads and writes during peak offer periods. Integration with MongoDB adds flexibility for analytics and reporting without affecting the transactional database.

The results demonstrate improved data consistency, faster redemption verification, and reliable audit trails for all financial operations.

13. Conclusion

The Gift Coupon Application effectively combines ACID-compliant relational databases with horizontally scalable architecture. It supports safe, concurrent transactions, ensuring data integrity across user, coupon, and payment entities.

Normalization up to 3NF minimizes redundancy and improves performance. By integrating MongoDB for analytics, the system provides flexibility without compromising the transactional core.

This database design can be extended to large-scale e-commerce systems, ensuring robust and fault-tolerant coupon management.

14. References

1. Oracle Database Documentation – Oracle Corporation

2. MongoDB Official Documentation

3. Silberschatz, Korth, Sudarshan – Database System Concepts

4. Raghu Ramakrishnan – Database Management Systems

5. CockroachDB and Google Spanner Architecture Whitepapers