

# Gaming Club App

## Abstract

This project develops a **Membership and Game Management System** using **Spring Boot and MongoDB** for managing the operations of a gaming center. The system enables administrators to manage **memberships, games, recharges, transactions, and daily collections** via secure REST APIs. Members are registered with an initial joining fee that updates their balance. Recharges and game plays are recorded as separate documents, and balances are automatically adjusted. The system maintains **daily collections** and provides detailed **member histories**. Authentication ensures only authorized administrators can access the system. With MongoDB's flexible schema and scalability, the solution eliminates manual tracking, improves accuracy, and offers a reliable digital platform for efficient gaming center management.

## Collections/Document Overview

| Collection Name | Fields (with notes)   |
|-----------------|---|
| members         | <code>_id ObjectId PK , name String NOT NULL , balance Double DEFAULT 0 , phone String UNIQUE</code>  |
| games           | <code>_id ObjectId PK , name String NOT NULL , price Double NOT NULL , description String</code>  |
| recharges       | <code>_id ObjectId PK , memberId ObjectId (ref → members._id) , amount Double NOT NULL , dateTime Date DEFAULT now()</code>                                     |
| transactions    | <code>_id ObjectId PK , memberId ObjectId (ref → members._id) , gameId ObjectId (ref → games._id) , amount Double NOT NULL , dateTime Date DEFAULT now()</code> |
| collections     | <code>_id ObjectId PK , date Date NOT NULL , amount Double NOT NULL</code>  |
| admin_users     | <code>_id ObjectId PK , username String UNIQUE NOT NULL , password String NOT NULL</code>   |

## MongoDB Collection Creation (Migration from SQL DDL)

In MongoDB you don't create schemas strictly (it's schema-less), but you can enforce structure with **JSON schema validation** at collection level. Below is the **equivalent of your SQL DDL** using `db.createCollection` with validators.

### 1. members

```
db.createCollection("members", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["name", "phone"],
      properties: {
        name: { bsonType: "string" },
        balance: { bsonType: "double" },
        phone: { bsonType: "string" }
      }
    }
  }
});
```

### 2. games

```
db.createCollection("games", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["name", "price"],
      properties: {
        name: { bsonType: "string" },
        price: { bsonType: "double" },
        description: { bsonType: "string" }
      }
    }
  }
});
```

---

### 3. recharges

```
db.createCollection("recharges", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["memberId", "amount"],
      properties: {
        memberId: { bsonType: "objectId" },
        amount: { bsonType: "double" },
        dateTime: { bsonType: "date" }
      }
    }
  }
});
```

---

### 4. transactions

```
db.createCollection("transactions", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["memberId", "gameId", "amount"],
      properties: {
        memberId: { bsonType: "objectId" },
        gameId: { bsonType: "objectId" },
        amount: { bsonType: "double" },
        dateTime: { bsonType: "date" }
      }
    }
  }
});
```

---

### 5. collections

```
db.createCollection("collections", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["date", "amount"],
      properties: {
        date: { bsonType: "date" },
        amount: { bsonType: "double" }
      }
    }
  }
});
```

6. admin\_users

```
db.createCollection("admin_users", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["username", "password"],
      properties: {
        username: { bsonType: "string" },
        password: { bsonType: "string" }
      }
    }
  }
});
```

Pages Design

1. Login Page

GAMING CLUB APP

Username

Password

Login

© 2025 Gaming Club. All rights reserved.

2. Membership Page

GAMING CLUB APP

Logo

[Membership]

[Member]

[Add Game]

[Collections]

Admin

Logout

CREATE MEMBERSHIP

Name

:

Phone

:

Membership Fee

:

[ Create Membership ]

© 2025 Gaming Club. All rights reserved.

3. Member Search Page

GAMING CLUB APP

Logo

[Membership]

[Member]

[Add Game]

[Collections]

Admin

Logout

MEMBER SEARCH

Phone

:

[ Search ]

© 2025 Gaming Club. All rights reserved.

4. Member Page (Tabbed Layout)

```
-----
|                                     |
| GAMING CLUB APP                    |
|-----|
| Logo | [Membership] [Member] [Add Game] [Collections] Admin |Logout|
|-----|
|                                     |
| MEMBER DETAILS                     |
|                                     |
| Name   : John Doe                 |
| Phone  : 9876543210               |
| Balance : ₹500                    |
|                                     |
| [ Games ] | [ Recharge History ] | [ Played Games ]
|-----|
```

>>> Games Tab (Default) <<<

List of Available Games:

| Name     | Price | Description      |               |
|----------|-------|------------------|---------------|
| Chess    | ₹50   | 2 players needed | [ Play Game ] |
| Carrom   | ₹100  | 2-4 players      | [ Play Game ] |
| Foosball | ₹150  | Multiple allowed | [ Play Game ] |

>>> Recharge History Tab <<<

| Date/Time        | Amount |
|------------------|--------|
| 2025-09-05 11:30 | ₹200   |
| 2025-09-02 15:10 | ₹300   |

>>> Played Games History Tab <<<

| Date/Time        | Game     | Amount |
|------------------|----------|--------|
| 2025-09-06 19:45 | Chess    | ₹50    |
| 2025-09-05 18:20 | Carrom   | ₹100   |
| 2025-09-02 16:00 | Foosball | ₹150   |

(Latest First Order)

## 5. Add Game Page

GAMING CLUB APP

Logo | [Membership] [Member] [Add Game] [Collections] Admin |Logout|

ADD GAME

Game Name : [\_\_\_\_\_]

Price (₹) : [\_\_\_\_\_]

Description : [\_\_\_\_\_]

Min Players : [\_\_\_] Multiple Allowed : [✓]

[ Add Game ]

© 2025 Gaming Club. All rights reserved.

6. Collections Page

GAMING CLUB APP

Logo | [Membership] [Member] [Add Game] [Collections] Admin |Logout|

Date: [\_\_\_\_\_] [ Search ] (Optional)

Recharge Collection on 07-09-2025

| Membr    | Recharge |
|----------|----------|
| Suman    | ₹50      |
| Sujan    | ₹100     |
| Rakshith | ₹150     |
| Total    | ₹300     |

© 2025 Gaming Club. All rights reserved.

🔗 API Endpoints with Request & Response

| Method | Endpoint | Request Body  | Sample Response Body   | Description   |
|--------|----------|---|--|---|
| POST   | /auth    | json {<br>"username":<br>"admin",<br>"password":<br>"admin" } | json "fake-jwt-token"  | Login<br>(currently<br>returns a<br>dummy<br>JWT token) |
| POST   | /members | json {<br>"name":<br>"John",<br>"phone":                      | json { "id": 1, "name": "John", "balance": 500.0, "phone":<br>"9876543210" } | Create new<br>membership                                |

| Method | Endpoint               | Request Body  | Sample Response Body   | Description   |
|--------|------------------------|---|--|---|
| POST   | /members/search        | <pre>json {   "phone":     "9876543210" }</pre>   | <pre>json { "member": { "id": 1, "name": "John", "phone": "9876543210", "balance": 450.0 }, "recharge_history": [ { "id": 1, "amount": 500.0, "dateTime": "2025-09-07T09:30:00" } ], "games": [ { "id": 2, "name": "Chess", "price": 50.0, "description": "Board game" } ], "played_history": [ { "id": 3, "date_time": "2025-09- 07T10:00:00", "game_name": "Chess", "amount": 50.0 } ] }</pre> | Search member (returns profile, recharges, available games, played history) |
| POST   | /game                  | <pre>json {   "name":     "Chess",   "price": 50,   "description":     "Board game" }</pre> | <pre>json { "id": 2, "name": "Chess", "price": 50.0, "description": "Board game" }</pre>   | Add new game  |
| POST   | /play                  | <pre>json {   "member_id":     1, "game_id":     2 }</pre>                                  | <pre>json "Game played successfully!" Or json "Insufficient balance!"</pre>  | Play game (deducts balance & creates transaction)                           |
| GET    | /collection/2025-09-07 | <i>(no body, path variable = date)</i>  | <pre>json [ { "transaction_id": 1, "member": "John", "recharge_amount": 500.0 } ]</pre>  | Get recharge transactions for a specific date                               |