

**International School**

**Capstone Project 2**

## CMU-SE 451 – C2SE.12

**Database Design**

**Version 1.1 Date: March 15th, 2021**

**Movie ticket management system**

**Submitted by Vo Anh Tuan, Nguyen Gia Hoa, Le Quang Trai, Vo Hoang Mai Khanh**

**Approved by**

**MSc Huy, Truong Dinh**

**Proposal Review Panel Representative:**

Name Signature Date

## Capstone Project 2- Mentor:

Name Signature Date

**PROJECT INFORMATION**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project acronym** | LET | | |
| **Project Title** | Learn English Together | | |
| **Start Date** | 26 Feb 2021 | **End Date** | 08 Jun 2021 |
| **Lead Institution** | International School, Duy Tan University | | |
| **Project Mentor** | MSc Huy, Truong Dinh | | |
| **Scrum master / Project Leader & contact details** | Ha, Le Thanh  Email: [lethanhhadtu@gmail.com](mailto:lethanhhadtu@gmail.com) Tel: 0334002818 | | |
| **Partner Organization** | Duy Tan University | | |
| **Project Web URL** |  | | |
| **Team members** | Name | Email | Tel |
| 1 | Vo Hoang Mai Khanh | [lethanhhadtu@gmail.com](mailto:lethanhhadtu@gmail.com) | 0334002818 |
| 2 | Vo Van Duc Tri | [xuanhieu.le.1999@gmail.com](mailto:xuanhieu.le.1999@gmail.com) | 0399706614 |
| 3 | Le Quang Trai | [ngongocmy851999@gmail.com](mailto:ngongocmy851999@gmail.com) | 0764497391 |
| 4 | Nguyen Gia Hoa |  |  |
| 5 | Vo Anh Tuan | [doanthong002@gmail.com](mailto:doanthong002@gmail.com) | 0886428208 |

# DOCUMENT APPROVALS

The following signatures are required for approval of this document.

|  |  |  |
| --- | --- | --- |
| Ha, Le Thanh  Student ID: 2321122516  *Scrum Master* | Signature | Date |
| Hieu, Le Xuan  Student ID: 2321124665  *Team Member* | Signature | Date |
| My, Ngo Ngoc  Student ID: 2321124970  *Team Member* | Signature | Date |
| Thong, Doan Trung Student ID: 2321124144  *Team Member* | Signature | Date |

**REVISION HISTORY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Comments** | **Author** | **Approval** |
| 1.0 | March 15th, 2021 | Initial Release | Ngoc My Ngo |  |
| 1.1 | April 20th, 2021 | Update | Ngoc My Ngo |  |

# TABLE OF CONTENT

1. [Introduction 5](#_bookmark0)
   1. [Purpose 5](#_bookmark1)
   2. [Scope 5](#_bookmark2)
   3. [Introduction about MongoDB 5](#_bookmark3)
2. [Database Diagram 6](#_bookmark4)
   1. [Table Overview 6](#_bookmark5)
   2. [Entity Relationship Diagram 7](#_bookmark6)
   3. [Table Relationship Diagram 8](#_bookmark7)
3. [Database Design for Sprint 9](#_bookmark8)
   1. [Table Group 9](#_bookmark9)
   2. [Table User 9](#_bookmark10)
   3. [Table Frame 10](#_bookmark11)
   4. [Table PrivateMessage 11](#_bookmark12)
   5. [Table PublicMessage 11](#_bookmark13)
   6. [Table Events 11](#_bookmark14)
   7. [Table Course 12](#_bookmark15)
   8. [Table CourseVocabulary 12](#_bookmark16)
   9. [Table CourseOfUsser 13](#_bookmark17)
   10. [Table Rating 13](#_bookmark18)
4. [Hardware and software Requirements 14](#_bookmark19)

## Introduction

This document describes the database design for a Cinema Ticket Management System. Its purpose is to ensure high performance during operations such as ticket booking, movie scheduling, user management, and complex data queries.

## Purpose

The purpose of the **Cinema Ticket Management System** is to provide an efficient, user-friendly, and automated solution for managing all ticket-related activities in a cinema. This includes the management of movies, showtimes, cinema rooms, seat arrangements, user bookings, and payment processes.

## Scope

The Database Cinema Ticket Management System has the following objectives:

* Manage users and login credentials.
* Manage movies, showtimes, and cinema rooms.
* Online ticket booking and payment processing.
* Admin panel for system administration.

## Introduction about MongoDB

MongoDB is a NoSQL database designed for scalability, performance, and high availability. Unlike traditional relational databases, MongoDB stores data in flexible, JSON-like documents, allowing it to handle complex and evolving data structures efficiently.

In the context of a movie ticket booking system, MongoDB provides several advantages:

* Flexible Schema Design: Entities such as users, movies, schedules, and bookings can be stored in collections with varying structures. For example, a booking document can embed user information and ticket details in a single record, making it easier to retrieve without performing complex joins.
* High Performance: With its document-based structure and support for indexing, MongoDB allows for fast read/write operations, which is essential for real-time features like seat selection, schedule browsing, and online ticket booking.
* Scalability: MongoDB supports horizontal scaling through sharding, which is useful when handling large volumes of data such as thousands of movie sessions, millions of users, and transactions.
* Built-in Replication: The use of replica sets ensures high availability and fault tolerance, which is crucial for a system that requires constant uptime.

{

user: " Nguyen Van A", ticketID: "T001",

movie: "Avengers: Endgame",  
cinema: "CGV Vincom",  
seat: D5",  
showTime: "2025-04-20T20:00:00",  
price: 95000,

}

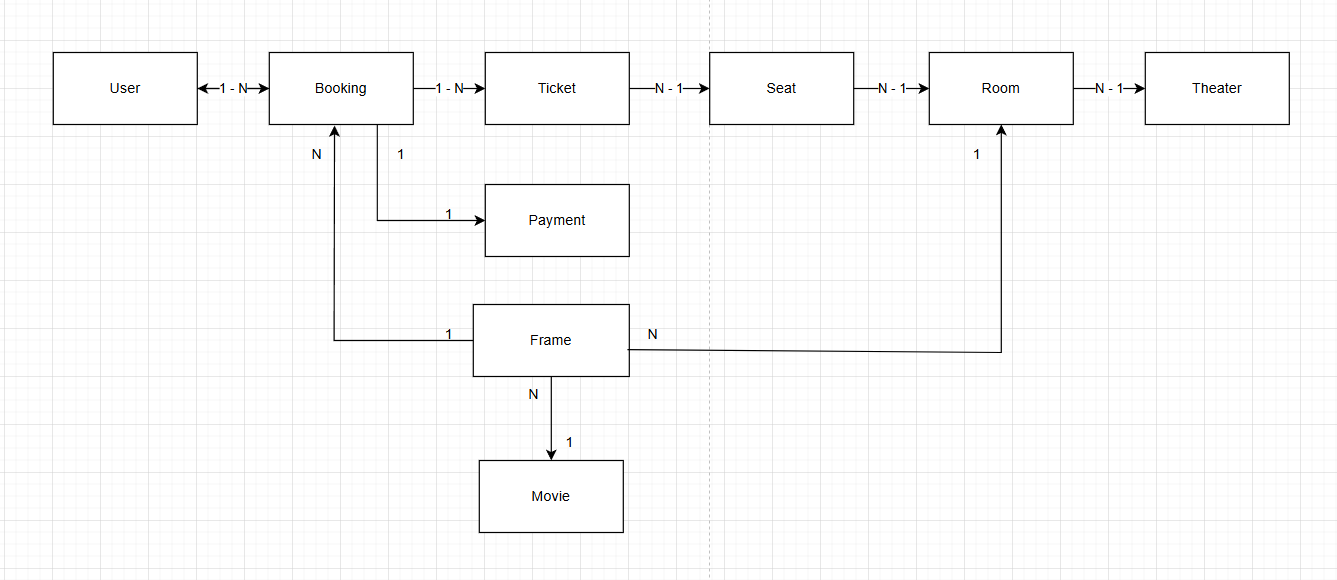
Overall, MongoDB is a powerful choice for building a movie ticket booking system due to its ability to manage diverse data models, handle high traffic, and scale with user demand. Its document-oriented nature aligns well with the application's need for agility and fast development.

## Database Diagram

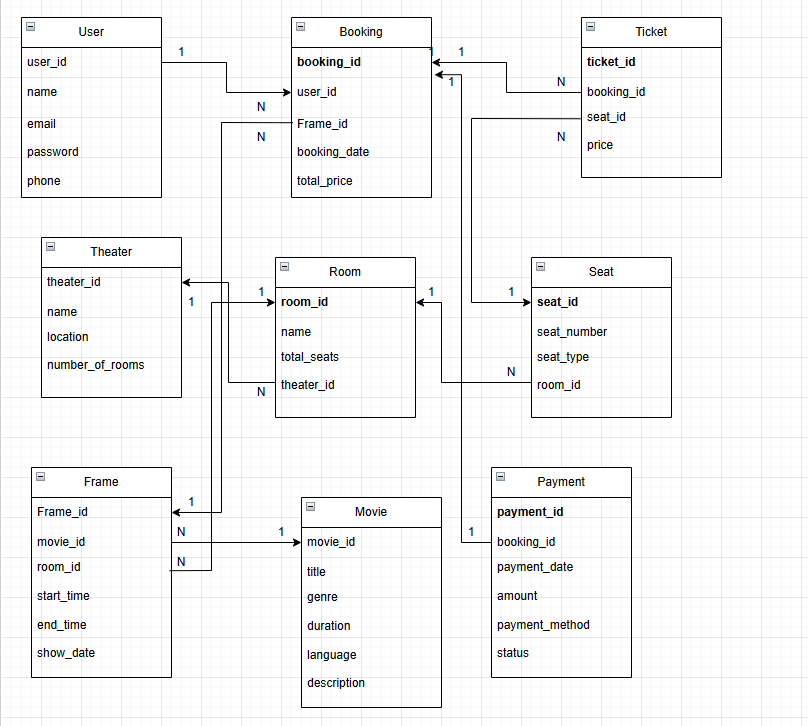
## Table Overview

|  |  |
| --- | --- |
| **Table Name** | **Short Description** |
| Group | Represents cinema staff groups or teams, such as customer service, ticketing, or operations teams. This table stores group names, roles, members, and related permissions. |
| User | Stores information about all users in the system, including customers and staff. It includes login credentials, contact details, roles, and account status. |
| Frame | Defines scheduling frameworks or time slots for movie screenings. It helps structure available time ranges in which showtimes can be configured. |
| PrivateMessage | Stores internal, private messages between staff members (e.g., between managers and ticket agents) for operational coordination. |
| PublicMessage | Contains announcements or public messages sent to users, such as movie promotions, event reminders, or cinema-wide notifications. |
| Quizz | Used for optional customer engagement—such as short quizzes or surveys about movies, cinema experiences, or promotional campaigns. |
| Event | Stores information about cinema-related events, including movie premieres, special screenings, fan meetings, or marketing events. Each event includes time, description, and organizer details. |

## Entity Relationship Diagram



## Table Relationship Diagram

****

## Database Design for Sprint

## Table Group

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | Group |
| Id | Field | Type | Constrain | Description |
| 1 | \_id | Object | Primary Key | Automatically generated ObjectId by MongoDB. |
| 2 | roomName | String | Not Null, Unique | Name of the cinema room. |
| 3 | roomCode | String | Unique | Unique code identifying the room. |
| 4 | managerId | Object | Foreign Key → User | ID of the staff managing this room. |
| 5 | seatLayoutId | Object | Foreign Key | Reference to seat layout configuration. |
| 6 | image | String | Optional | Image representing the cinema room. |
| 7 | scheduleInfo | String | Optional | Notes or schedule metadata for the room. |
| 8 | videoSystem | date | Optional | Description of video system. |
| 9 | audioSystem | String | Optional | Type of audio system used. |
| 10 | isActive | Boolean | Default: true | Room status (active/inactive). |
| 11 | createdAt | Date | Default: now | Timestamp when the room was added to the system. |
| 12 | updatedAt | Date | Updated on changes | Last modification timestamp. |
| 13 | createdBy | Object | Foreign Key → User | ID of the user who created this room record. |

## Table User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User | | | | |
| Id | Field | Type | Constrain | Description |
| 1 | \_id | Object | Primary key | Auto-generated unique ID by MongoDB. |
| 2 | userName | String | Not Null | Display name of the user. |
| 3 | password | String | Encrypted, Not Null | Encrypted password for login. |
| 4 | email | String | Unique, Not Null | User's email address (used for login and password recovery). |
| 5 | phoneNumber | String | Optional | User’s phone number. |
| 6 | dob | Date | Optional | Date of birth. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7 | role | String | Default: "customer" | User role: admin, staff, or customer. |
| 8 | isActive | Boolean | Default: true | Account status (active/inactive). |
| 9 | avatar | String | Optional | URL or path to user avatar/profile picture. |
| 10 | registeredAt | String | Default: now | Date and time the account was created. |
| 11 | lastLogin | Date | Optional | Timestamp of the user’s most recent login. |
| 12 | address | String | Optional | User’s home or billing address. |
| 13 | bookings | Object | Foreign Key → Ticket | List of ticket IDs this user has booked. |

## Table Frame

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Frame | | | | |
| Id | Field | Type | Constrain | Description |
| 1 | \_id | Object | Primary key | Auto-generated unique ID for each showtime. |
| 2 | movieId | Object | Foreign Key → Movie | ID of the movie being shown. |
| 3 | roomId | Object | Foreign Key → Room | ID of the cinema room where the movie is screened. |
| 4 | startTime | Date | Not Null | Date and time the screening starts. |
| 5 | endTime | Date | Not Null | Date and time the screening ends. |
| 6 | price | Number | Not Null | Ticket price for this showtime. |
| 7 | isAvailable | Boolean | Default: true | Status indicating if the showtime is currently open for booking. |
| 8 | createdBy | Object | Foreign Key → User | ID of the staff/admin who created the showtime. |
| 9 | createdAt | Date | Default: now | Timestamp when the showtime was created. |
| 10 | updatedAt | Date | Optional | Timestamp when the showtime was last updated. |

## Table PrivateMessage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PrivateMessage | | | | |
| Id | Field | Type | Constrain | Description |
| 1 | \_id | Object | Primary key | Auto-generated unique ID for the message. |
| 2 | senderId | Object | Foreign Key → User | ID of the user who sends the message (customer or staff). |
| 3 | receiverId | Object | Foreign Key → User | ID of the user who receives the message. |
| 4 | message | String | Not Null | Content of the private message. |
| 5 | createdAt | Number | Default: now | Timestamp when the message was created. |
| 6 | sentAt | Date | Optional | Datetime when the message was sent. |
| 7 | isRead | Boolean | Default: false | Whether the receiver has read the message or not. |
| 8 | ticketRefId | Object | Optional | Reference to a specific ticket if the message is about a booking. |

## Table PublicMessage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PublicMessage | | | | |
| Id | Field | Type | Constrain | Description |
| 1 | \_id | Object | Primary Key | Auto-generated unique ID for the message. |
| 2 | authorId | Object | Foreign Key → User | ID of the user who created the message. |
| 3 | targetGroup | String | Optional | Target group of users. |
| 4 | title | String | Not Null | Title or subject of the message. |
| 5 | message | String | Not Null | Content of the public message. |
| 6 | createdAt | Number | Default: now | Timestamp when the message was created. |
| 7 | publishedAt | Date | Optional | Optional Human-readable date when the message was published. |
| 8 | isPinned | Boolean | Default: false | Whether the message is pinned or highlighted on the homepage. |
| 9 | status | String | Default: "active" | Status of the message: "active", "draft", "archived". |

## Table Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Event | | | | |
| Id | Field | Type | Constrain | Description |
| 1 | \_id | Object | Primary key | Auto-generated ID for the event. |
| 2 | eventName | String | Not Null | Name of the event. |
| 3 | description | String | Optional | Description or details about the event. |
| 4 | bannerImage | String | Optional | Path or URL to event promotional image/banner. |
| 5 | location | String | Optional | Location of the event |
| 6 | startTime | Date | Not Null | Start time and date of the event. |
| 7 | endTime | Date | Optional | End time and date of the event. |
| 8 | createdBy | Object | Foreign Key → User | ID of staff or admin who created the event. |
| 9 | createdAt | Number | Default: now | Timestamp when the event record was created. |
| 10 | updatedAt | Date | Optional | Last time the event information was updated. |
| 11 | isActive | Boolean | Default: true | Indicates whether the event is currently active or expired. |

## Table Course

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Course | | | | |
| Id | Field | Type | Constrain | Description |
| 1 | \_id | Object | Primary key | MongoDB auto-generates an ObjectId. Unique identifier for movie. |
| 2 | nameMovie | String | Not Null | Title of the movie. |
| 3 | description | String | Optional | Summary or plot of the movie. |
| 4 | genre | String | Optional | Genre of the movie |
| 5 | language | String | Optional | Language of the movie. |
| 6 | duration | Number | Not Null | Duration of the movie in minutes. |
| 7 | director | String | Optional | Director of the movie. |
| 8 | cast | String | Optional | List of main actors. |
| 9 | poster | String | Optional | URL to the movie poster image. |
| 10 | trailerLink | String | Optional | URL to the trailer video |
| 11 | releaseDate | Date | Optional | Official release date. |
| 12 | userCreate | Object | Foreign Key → User | ID of the staff who added the movie. |
| 13 | createdAt | Number | Default: now | Timestamp when the movie was added. |
| 14 | updatedAt | Date | Optional | Timestamp when the movie was last updated. |

## Table CourseVocabulary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CourseVocabulary | | | | |
| Id | Field | Type | Constrain | Description |
| 1 | \_id | Object | Primary key | MongoDB auto-generates an ObjectId. Unique identifier for each tag. |
| 2 | movieId | Object | Foreign Key → Movie | ID of the movie this tag is associated with. |
| 3 | tagName | String | Not Null | Tag name or keyword |
| 4 | type | String | Optional | Type of tag |
| 5 | description | String | Optional | Additional notes or explanation of the tag. |
| 6 | iconImage | String | Optional | Optional icon or image related to the tag. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7 | userCreate | Object | Foreign Key → User | ID of the user who created this tag entry. |
| 8 | createdAt | Number | Default: now | Timestamp when the tag was created. |
| 9 | updatedAt | Date | Optional | Last update time of the tag. |

## Table CourseOfUsser

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CourseOfUsser | | | | |
| Id | Field | Type | Constrain | Description |
| 1 | \_id | Object | Primary key | MongoDB auto-generates an ObjectId. Unique identifier for the ticket. |
| 2 | showtimeId | Object | Foreign Key → Showtime | ID of the showtime the ticket is associated with. |
| 3 | userId | Object | Foreign Key → User | ID of the user who booked the ticket. |
| 4 | seatId | Object | Foreign Key → Seat | ID of the selected seat in the room. |
| 5 | bookingTime | Default: now | Default: now | Timestamp when the ticket was booked. |
| 6 | updatedAt | Date | Optional | Timestamp of the last update to the ticket. |
| 7 | price | Number | Not Null | Price of the ticket at the time of booking. |
| 8 | status | String | Default: "pending" | Booking status: "pending", "paid", "cancelled". |
| 9 | qrCode | String | Optional | QR code string used for check-in at the cinema. |

## Table Rating

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rating | | | | |
| Id | Field | Type | Constrain | Description |
| 1 | \_id | Object | Primary key | MongoDB auto-generates an ObjectId. Unique ID for each rating. |
| 2 | userId | Object | Foreign Key → User | ID of the user who gave the rating. |
| 3 | movieId | Object | Foreign Key → Movie | ID of the movie being rated. |
| 4 | star | Number | Range: 1–5 | Star rating given by the user. |
| 5 | comment | String | Optional | Text review or comment provided by the user. |
| 6 | createdAt | Nummber | Default: now | Timestamp when the rating was created. |
| 7 | updatedAt | Date | Optional | Timestamp of the last update to the rating. |
| 8 | isPublic | Boolean | Default: true | Whether the review is publicly visible to other users. |

## Hardware and software Requirements

This section outlines the required technologies and environment to support the cinema ticket management system:

|  |  |
| --- | --- |
| **Attributes of Easy English WEBSITE** | |
| **Attributes** | **Descriptions** |
| **Database** | MongoDB |
| **Software** | Reactjs, Nodejs, Polling, Elasticsearch, Redis, Websocket, Mail gun |
| **Hardware** | Computer |
| **Library** | Hook, Redux, React hook form, Bootstrap 4, Fontawesome, Material ui, Express.js. |