Thiết kế dữ liệu LingoLeap

Version 1.0

Sinh viên thực hiện:

21810203 – Nguyễn Đức Cang

21810220 – Hồng Phương Tuấn

21810221 – Phan Thanh Tuấn **Bảng ghi nhận thay đổi tài liệu**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ngày** | **Phiên bản** | **Mô tả** | **Người thay đổi** |
| 16/12/2024 | 1.0 | Khởi tạo thiết kế cơ sở dữ liệu | 21810220 – Hồng Phương Tuấn |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Mục lục

[*1. Tổng Quan* 4](#_Toc185245610)

[*1.1 Công nghệ sử dụng* 4](#_Toc185245611)

[*1.2 Đặc điểm chính* 4](#_Toc185245612)

[*1.3 Các collections chính* 4](#_Toc185245613)

[*2. Chi tiết các collections* 5](#_Toc185245614)

[*2.1 Các Collections của User* 5](#_Toc185245615)

[*2.2 Các Collections của Test* 10](#_Toc185245616)

[*3. Quan hệ giữa các collections* 18](#_Toc185245617)

[*3.1 Core collections và mối quan hệ* 19](#_Toc185245618)

[*3.2 Các mối quan hệ phức tạp* 21](#_Toc185245619)

[*3.3 Indexing Strategy* 21](#_Toc185245620)

[*3.4 Practices* 22](#_Toc185245621)

***Tài Liệu Cơ Sở Dữ Liệu Hệ Thống Thi Trực Tuyến***

*1. Tổng Quan*

*1.1 Công nghệ sử dụng*

* *Database Engine: MongoDB*
* *Schema Management: Mongoose*
* *Type: NoSQL Document Database*
* *Version: MongoDB 5.0+*

*1.2 Đặc điểm chính*

* *Lưu trữ dữ liệu dạng document (BSON)*
* *Hỗ trợ schema linh hoạt*
* *Quan hệ thông qua references*
* *Hỗ trợ indexing đa dạng*
* *Tích hợp với Node.js tốt*

*1.3 Các collections chính*

*1.3.1 User and Profile*

* *users*
* *userProfiles*
* *userStatistics*
* *loginSessions*
* *subscriptions*

*1.3.2 Testing System*

* *tests*
* *questions*
* *testAttempts*
* *testResults*
* *reviews*

*2. Chi tiết các collections*

*2.1 Các Collections của User*

*2.1.1 User Collection*

|  |  |  |
| --- | --- | --- |
| ***Field Name*** | ***Type*** | ***Description*** |
| \_id | ObjectId | Auto-generated by MongoDB |
| email | String | Required, unique, lowercase |
| username | String | Required, unique |
| password | String | Required, minimum length: 8 |
| dateOfBirth | Date | Date of birth of the user |
| phoneNumber | String | User's phone number |
| firstName | String | First name of the user |
| lastName | String | Last name of the user |
| address | Object | Includes street, city, country fields |
| avatar | String | URL of the user's avatar image |
| isVerified | Boolean | True if the user's account is verified |
| isDeleted | Boolean | True if the user's account is marked as deleted |
| role | String | Enum: user, teacher, admin (default: user) |
| userProfile | ObjectId | References the user's profile |
| userStatistics | ObjectId | References the user's statistics |
| createdAt | Date | Account creation date |
| updatedAt | Date | Account last updated date |

*Indexes:*

* *email: Unique index for user emails.*
* *username: Unique index for usernames.*
* *role: Index for filtering users by role.*

*2.1.2 UserProfile Collection*

|  |  |  |
| --- | --- | --- |
| ***Field Name*** | ***Type*** | ***Description*** |
| \_id | ObjectId | Auto-generated by MongoDB |
| user | ObjectId | Required, unique, references User |
| testTaken | [ObjectId] |  |
| testScore | [Number] |  |
| achievements | [String] |  |
| preferences | Object | Includes notificationSettings and studyReminders |
| learningGoals | [String] |  |
| targetExams | [String] |  |
| studyPlan | ObjectId |  |
| progressTracking | Object | Includes progress metrics such as vocabularyMastered and grammarPointsLearned |
| deviceInfo | [{ deviceType: String, lastUsed: Date }] |  |
| socialLinks | Object | Includes twitter, linkedin, github |
| socialConnections | Object | Includes studyBuddies, following, and followers |
| feedbackProvided | [ObjectId] |  |
| createdAt | Date |  |
| updatedAt | Date |  |
|  |  |  |

*Indexes:*

* *user: Unique*
* *socialConnections.studyBuddies: Non-unique*

*2.1.3 User Statistic Collection*

|  |  |  |
| --- | --- | --- |
| ***Field Name*** | ***Type*** | ***Description*** |
| \_id | ObjectId | Auto-generated by MongoDB |
| user | ObjectId | Required, unique, references User |
| testsCompleted | Object | Includes total tests and breakdown by difficulty and type |
| averageScores | Object | Includes overall average and section-wise scores (reading, listening, writing, speaking) |
| questionStats | Object | Includes total attempted, correct answers, and stats by question type (single\_choice, multiple\_choice, fill\_in\_the\_blank) |
| timeStats | Object | Includes totalTimeSpent, averageTestDuration, and averageQuestionTime |
| progressOverTime | [{ date: Date, score: Number, testId: ObjectId }] | List of progress entries over time |
| lastUpdated | Date |  |
| createdAt | Date |  |
| updatedAt | Date |  |

*Indexes:*

* *user: Unique*

*2.1.4 LoginSession Collection*

|  |  |  |
| --- | --- | --- |
| ***Field Name*** | ***Type*** | ***Description*** |
| \_id | ObjectId | Auto-generated by MongoDB |
| user | ObjectId | Required, references User |
| deviceInfo | String | Information about the user’s device |
| ipAddress | String | User's IP address |
| refreshToken | String | Required, stores the refresh token |
| isActive | Boolean | Defaults to true, indicates status |
| lastActivity | Date | Last recorded activity timestamp |
| expiresAt | Date | Expiry date of the refresh token |
| refreshTokenUsed | Boolean | Indicates whether the token was used |
| clientId | String | Identifier for the client application |
| createdAt | Date | Record creation timestamp |
| updatedAt | Date | Record update timestamp |

*Indexes:*

* *User*

*2.1.5 Subscriptions Collection*

|  |  |  |
| --- | --- | --- |
| ***Field Name*** | ***Type*** | ***Description*** |
| \_id | ObjectId | Auto-generated by MongoDB |
| user | ObjectId | Required, references User, unique |
| plan | Object | Contains subscription plan details |
| plan.type | String | Enum: ["free", "basic", "premium"] |
| plan.expiryDate | Date | Expiry date of the current plan |
| paymentHistory | Array | List of payment records |
| paymentHistory[].amount | Number | Payment amount |
| paymentHistory[].date | Date | Payment date |
| paymentHistory[].planType | String | Plan type associated with the payment |
| isActive | Boolean | Indicates if the subscription is active |
| createdAt | Date | Record creation timestamp |
| updatedAt | Date | Record update timestamp |

*Indexes*

* *user: 1 (unique)*
* *plan.type: 1*
* *isActive: 1*

*2.2 Các Collections của Test*

*2.2.1 Test Collection*

|  |  |  |
| --- | --- | --- |
| ***Field Name*** | ***Type*** | ***Description*** |
| \_id | ObjectId | Auto-generated by MongoDB |
| title | String | Required, trimmed, max 100 characters |
| description | String | Optional, max 500 characters |
| duration | Number | Required, calculated as the sum of all section durations; must match sum of section durations, min 1 |
| sections | Array of Objects | Each section includes name, instructions, sectionScore, questions, passages, and duration |
| sections.name | String | Enum: sections, required |
| sections.instructions | String | Optional, max 500 characters |
| sections.sectionScore | Number | Required, min 0 |
| sections.questions | Array of Objects | Optional for reading and listening; required otherwise. Each question must have \_id and points |
| sections.duration | Number | Required, min 1 |
| sections.passages | Array of Objects | Required for reading and listening sections; contains \_id (reference to BasePassage) and points |
| totalPossibleScore | Number | Required, min 0 |
| passingScore | Number | Required, min 0; cannot exceed totalPossibleScore |
| difficulty | String | Enum: difficultyLevels, required |
| testType | String | Enum: testTypes, required |
| createdBy | ObjectId | Reference to User, required |
| isPublished | Boolean | Default: true |
| attemptsAllowed | Number | Default: 1, min 1 |
| availableFrom | Date | Required, default: current date |
| availableUntil | Date | Required, default: 365 days from current date |
| averageRating | Number | Default: 0, min 0, max 5 |
| totalReviews | Number | Default: 0 |
| participantCount | Number | Default: 0, min 0 |
| Timestamps |  | createdAt, updatedAt fields automatically added |

*Indexes:*

* *\_id: 1*
* *createdBy: 1*

*2.2.2 TestAttempt Collection*

|  |  |  |
| --- | --- | --- |
| ***Field Name*** | ***Type*** | ***Description*** |
| \_id | ObjectId | Auto-generated by MongoDB |
| user | ObjectId | Reference to User, required |
| test | ObjectId | Reference to Test, required |
| startTime | Date | Required, default: current timestamp |
| endTime | Date | Optional, marks the completion time of the attempt |
| maxEndTime | Date | Required, represents the deadline for completing the test |
| status | String | Enum: ["in-progress", "completed", "abandoned"], default: "in-progress", required |
| result | ObjectId | Optional, reference to a TestResult document |
| totalScore | Number | Default: 0, min: 0 |
| answers | Array of Objects | Contains answers submitted during the attempt |
| answers.question | ObjectId | Reference to a Questions document |
| answers.answer | Mixed | Stores the user’s answer, adaptable for various question types (e.g., text, arrays for multiple-choice, etc.) |
| Timestamps | N/A | Automatically adds createdAt and updatedAt fields |

*Indexes:*

* *User: 1*
* *Tes: 1*

*2.2.3 TestResult Collection*

|  |  |  |
| --- | --- | --- |
| ***Field Name*** | ***Type*** | ***Description*** |
| \_id | ObjectId | Auto-generated by MongoDB |
| \_id | ObjectId | Unique identifier for the document. |
| testAttempt | ObjectId | Reference to the TestAttempt document, required and unique. |
| user | ObjectId | Reference to the User document. |
| test | ObjectId | Reference to the Test document. |
| totalScore | Number | Total score achieved in the test, required, cannot be negative. |
| maxTotalScore | Number | Maximum possible score for the test, defaults to 100, cannot be negative. |
| scorePercentage | Number | Percentage score, must be between 0 and 100. |
| totalQuestions | Number | Total number of questions in the test, required, must be at least 1. |
| sectionScores | Array | Array of sectionScoreSchema documents. |
| questionScores | Array | Array of questionScoreSchema documents. |
| writingQuestionResults | Array | Array of writingQuestionResultSchema documents. |
| speakingResult | Object | Object following SpeakingResultSchema. |
| feedback | String | Feedback for the test result. |
| gradedBy | ObjectId | Reference to the User document who graded the test. |
| gradedAt | Date | Date and time when the test was graded. |
| createdAt | Date | Timestamp when the document was created. |
| updatedAt | Date | Timestamp when the document was last updated. |

*Indexes:*

* *User: 1*
* *Test: 1*
* *TestAttemp: 1*

***sectionScoreSchema***

|  |  |  |
| --- | --- | --- |
| ***Field Name*** | ***Type*** | ***Description*** |
| sectionType | String | Section type (reading, listening, writing, speaking, general). |
| score | Number | Score for the section, required, cannot be negative. |
| totalQuestions | Number | Total questions in the section, required, at least 1. |

*questionScoreSchema*

|  |  |  |
| --- | --- | --- |
| ***Field Name*** | ***Type*** | ***Description*** |
| \_id | ObjectId | Reference to the BaseQuestion document, required. |
| score | Number | Score for the question, required, cannot be negative. |

*aspectSchema*

|  |  |  |
| --- | --- | --- |
| ***Field Name*** | ***Type*** | ***Description*** |
| aspect | String | Aspect being scored, required. |
| score | Number | Score for the aspect, between 0 and 10, required. |
| feedback | String | Feedback for the aspect, required. |

*writingQuestionResultSchema*

|  |  |  |
| --- | --- | --- |
| ***Field Name*** | ***Type*** | ***Description*** |
| question | ObjectId | Reference to the BaseQuestion document, required. |
| aspects | Array | Array of aspectSchema documents, required. |
| summary\_feedback | String | Summary feedback for the question, required. |
| totalScore | Number | Total score for the question, required. |

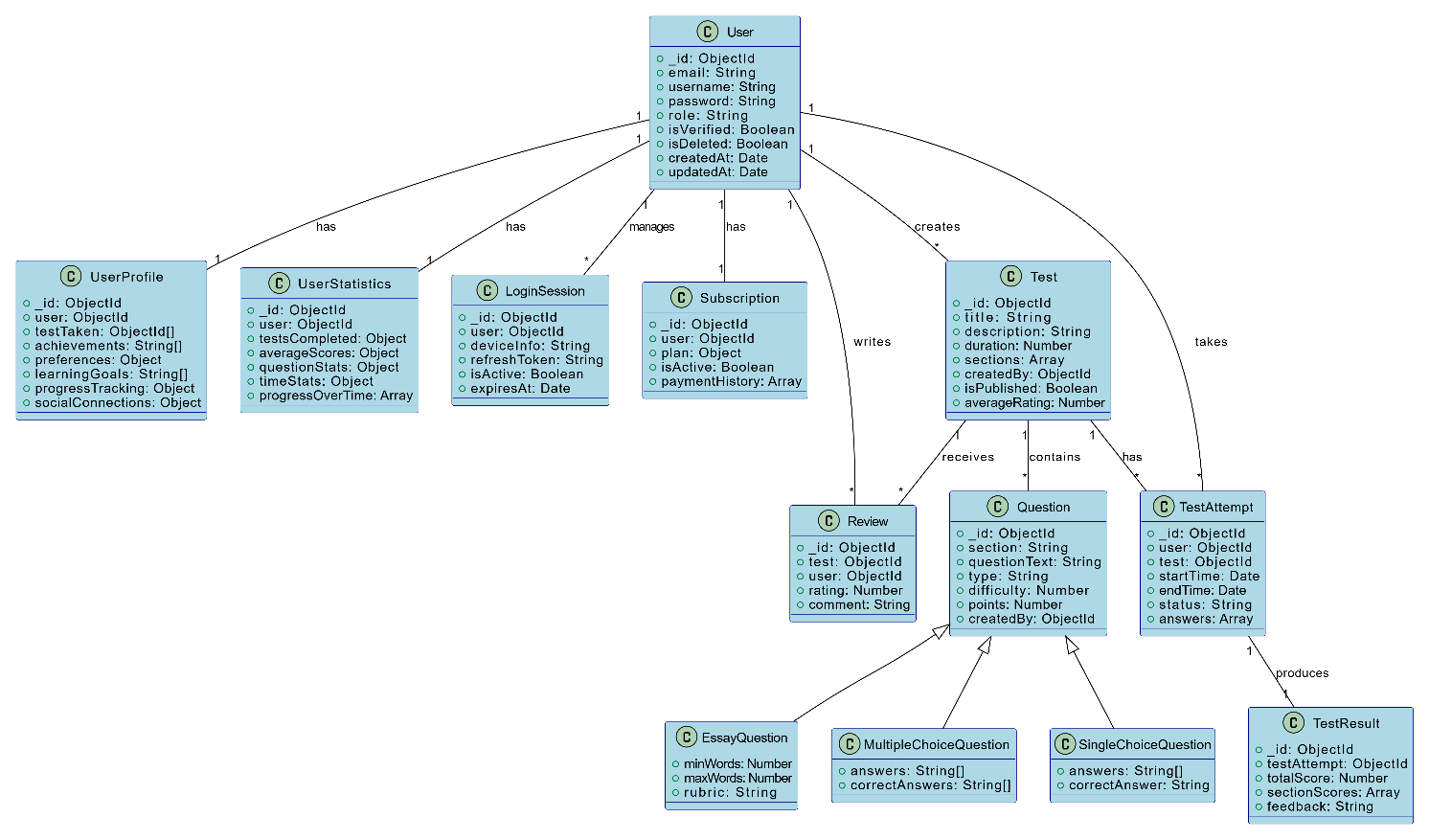
*SpeakingResultSchema*

|  |  |  |
| --- | --- | --- |
| ***Field Name*** | ***Type*** | ***Description*** |
| fluency | Number | Score for fluency. |
| pronunciation | Number | Score for pronunciation. |
| vocabulary | Number | Score for vocabulary. |
| overallCommunication | Number | Overall communication score. |
| totalScore | Number | Total score for the speaking section. |
| feedback | String | Feedback for the speaking result. |

*2.2.4 Review Collection*

|  |  |  |
| --- | --- | --- |
| ***Field Name*** | ***Type*** | ***Description*** |
| \_id | ObjectId | Auto-generated by MongoDB |
| test | ObjectId | Reference to the Test document, required. |
| user | ObjectId | Reference to the User document, required. |
| rating.value | Number | Rating value between 1 and 5. |
| rating.lastUpdated | Date | Timestamp for when the rating was last updated, defaults to now. |
| comments | Array | Array of comments associated with the review. |
| comments.text | String | Comment text, required, maximum 500 characters. |
| comments.createdAt | Date | Timestamp for when the comment was created, defaults to now. |
| createdAt | Date | Timestamp when the document was created. |
| updatedAt | Date | Timestamp when the document was last updated. |

*3. Quan hệ giữa các collections*

**

*3.1 Core collections và mối quan hệ*

*3.1.1 User-Center Collection*

* *User - UserProfile (1:1)*
* *Mỗi User có một UserProfile duy nhất*
* *Reference hai chiều để truy vấn hiệu quả*
* *Cascade deletion khi xóa User*
* *User - UserStatistics (1:1)*
* *Theo dõi thống kê học tập của user*
* *Cập nhật real-time khi có hoạt động mới*
* *Tự động tạo khi tạo User mới*
* *User - Subscription (1:1)*
* *Quản lý gói đăng ký của user*
* *Theo dõi lịch sử thanh toán*
* *Kiểm tra quyền truy cập tính năng*
* *User - LoginSession (1:N)*
* *Quản lý nhiều phiên đăng nhập*
* *Theo dõi thiết bị và IP*
* *Hỗ trợ refresh token*

*3.1.2 Test-related Collection*

* *Test - Question (1:N)*
* *Test chứa nhiều Question*
* *Questions có thể được tái sử dụng*
* *Phân loại theo sections*
* *Test - TestAttempt (1:N)*
* *Một Test có nhiều lần thử*
* *Mỗi TestAttempt liên kết với một User*
* *Theo dõi tiến trình và thời gian*
* *TestAttempt - TestResult (1:1)*
* *Mỗi lần thử có một kết quả*
* *Lưu trữ chi tiết câu trả lời*
* *Tính điểm và phân tích*
* *Test - Review (1:N)*
* *Users có thể đánh giá Test*
* *Tính điểm trung bình rating*
* *Lưu trữ comments và feedback*

*3.1.3 Question Hierarchy*

* *BaseQuestion --> MultipleChoice*
* *BaseQuestion --> SingleChoice*
* *BaseQuestion --> TrueFalse*
* *BaseQuestion --> FillInBlank*
* *BaseQuestion --> Matching*
* *BaseQuestion --> OpenEnded*

*3.2 Các mối quan hệ phức tạp*

*3.2.1 Cascading Operation*

* *User Deleteion*
* *Delete UserProfile*
* *Delete UserStatistics*
* *Deactivate Subscription*
* *Remove LoginSessions*
* *Archive TestAttempts*
* *Test Deletion*
* *Archive TestAttempts*
* *Remove from UserProfiles*
* *Delete Reviews*
* *Update Statistics*

*3.2.2 Cross-Collection Updates*

* *Complete test Attempt*
* *Cập nhật TestAttempt status*
* *Tạo TestResult*
* *Cập nhật UserStatistics*
* *Cập nhật Test.participantCount*
* *Thêm vào UserProfile.testTaken*

*3.3 Indexing Strategy*

*3.3.1 Primary Indexes*

* *User-related*
* *users: { email: 1, username: 1 }*
* *userProfiles: { user: 1 }*
* *userStatistics: { user: 1 }*
* *Test-related*
* *testAttempts: { user: 1, test: 1 }*
* *reviews: { test: 1, user: 1 }*

*3.4 Practices*

* *Denormalization Strategy*
* *Lưu trữ thông tin thường xuyên truy cập*
* *Cập nhật đồng bộ khi có thay đổi*
* *Cân bằng giữa performance và consistency*
* *Transaction Management*
* *Sử dụng sessions cho operations phức tạp*
* *Đảm bảo tính nhất quán của dữ liệu*
* *Rollback khi có lỗi*
* *Data Integrity*
* *Kiểm tra references trước khi xóa*
* *Cập nhật cascade khi cần thiết*
* *Validate dữ liệu trước khi lưu*
* *Query Optimization*
* *Sử dụng indexes hiệu quả*
* *Giới hạn số lượng references*
* *Tối ưu hóa aggregation pipelines*