Hồ sơ Phân tích LingoLeap

Version 1.0

Sinh viên thực hiện:

21810203 – Nguyễn Đức Cang

**Bảng ghi nhận thay đổi tài liệu**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ngày** | **Phiên bản** | **Mô tả** | **Tác giả** |
| 15/12/2024 | 1.0 | Khởi tạo hồ sơ phân tích | Nguyễn Đức Cang |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Mục lục

[1. *Sơ đồ* *lớp* 4](#_Toc185246994)

[*1.1* *Sơ đồ lớp* 4](#_Toc185246995)

[*1.2* *Danh sách các lớp đối tượng* 10](#_Toc185246996)

[*1.3* *Mô tả chi tiết* 14](#_Toc185246997)

[*2.* *Sơ đồ trạng thái* 24](#_Toc185247000)

[2.1 Các Trạng Thái Chính 25](#_Toc185247001)

[2.2 Các Chuyển Tiếp (Transitions) 26](#_Toc185247002)

[2.3 Xử Lý Đặc Biệt 27](#_Toc185247003)

# *Sơ đồ* *lớp*

* 1. *Sơ đồ lớp*
     1. *Lớp User*

A screenshot of a computer

Description automatically generated

* + 1. *Lớp UserProfile*

A screenshot of a computer program

Description automatically generated

* + 1. *Lớp UserStatistic*

A screenshot of a computer program

Description automatically generated

* + 1. *Lớp Test*

A screenshot of a computer

Description automatically generated

* + 1. *Lớp TestAttempt*

A screenshot of a computer

Description automatically generated

* + 1. *Lớp TestResult*

A screenshot of a test results

Description automatically generated

* + 1. *Lớp Questions*

A screenshot of a computer

Description automatically generated

* + - 1. *SingleChoice*

A diagram of a question

Description automatically generated

* + - 1. *Multiple Choice*

A screenshot of a computer

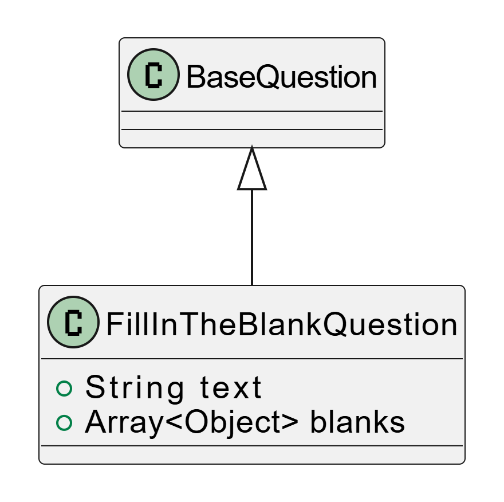
Description automatically generated

* + - 1. *TrueFalse*

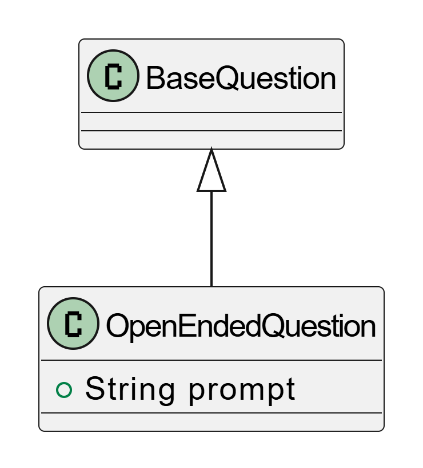
A diagram of a question

Description automatically generated

* + - 1. *Fill in the blank*



* + - 1. *Open-Ended*



* + - 1. *Essay*

A screenshot of a computer

Description automatically generated

* + - 1. *Matching*

A screenshot of a computer

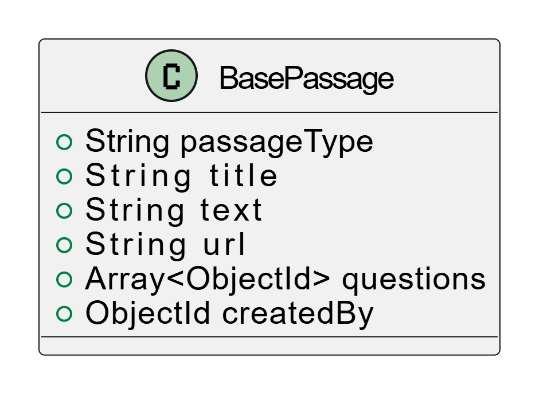
Description automatically generated

* + - 1. *Ordering*

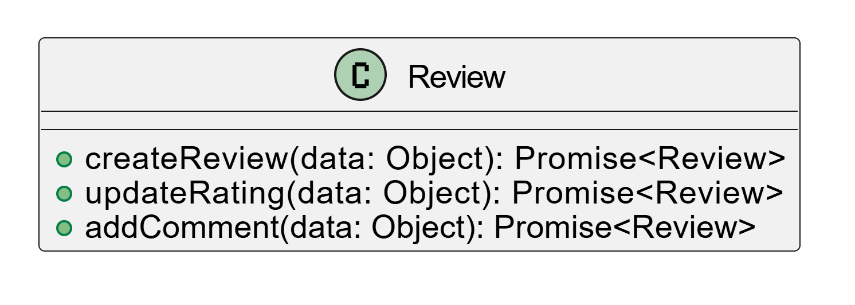
A diagram of a computer

Description automatically generated with medium confidence

* + - 1. *Passages*

**

* + 1. *Lớp Review*



* 1. *Danh sách các lớp đối tượng*

***1.2.1 Tổng quan các classes***

***Core Classes***

1. *User*

* *Quản lý thông tin người dùng*
* *Xử lý authentication và authorization*
* *Lưu trữ thông tin cá nhân cơ bản*

1. *Userprofile*

* *Mở rộng thông tin chi tiết của người dùng*
* *Theo dõi tiến trình học tập*
* *Lưu trữ preferences và thông tin social*

1. *UserStatistic*

* *Theo dõi số liệu thống kê của người dùng*
* *Lưu trữ kết quả các bài test*
* *Phân tích hiệu suất học tập*

***TesingClasses***

1. *Test*

* *Quản lý cấu trúc bài kiểm tra*
* *Định nghĩa các sections và scoring rules*
* *Theo dõi thời gian và điều kiện làm bài*

1. *TestAttempt*

* *Quản lý một lần thử làm bài test*
* *Theo dõi câu trả lời và thời gian*
* *Tính toán điểm số*

1. *TestResult*

* *Lưu trữ kết quả chi tiết của bài test*
* *Phân tích điểm theo từng phần*
* *Cung cấp feedback*

***Question Classes***

1. *BaseQuestion*

*Class cơ sở cho tất cả các loại câu hỏi*

*Định nghĩa các thuộc tính chung*

1. *Các loại Question (extends BaseQuestion)*

* *MultipleChoiceQuestion: Câu hỏi nhiều lựa chọn*
* *SingleChoiceQuestion: Câu hỏi một lựa chọn*
* *TrueFalseQuestion: Câu hỏi đúng/sai*
* *FillInTheBlankQuestion: Câu hỏi điền vào chỗ trống*
* *MatchingQuestion: Câu hỏi nối*
* *OrderingQuestion: Câu hỏi sắp xếp thứ tự*
* *OpenEndedQuestion: Câu hỏi mở*
* *EssayQuestion: Câu hỏi luận*

*PassageClassses*

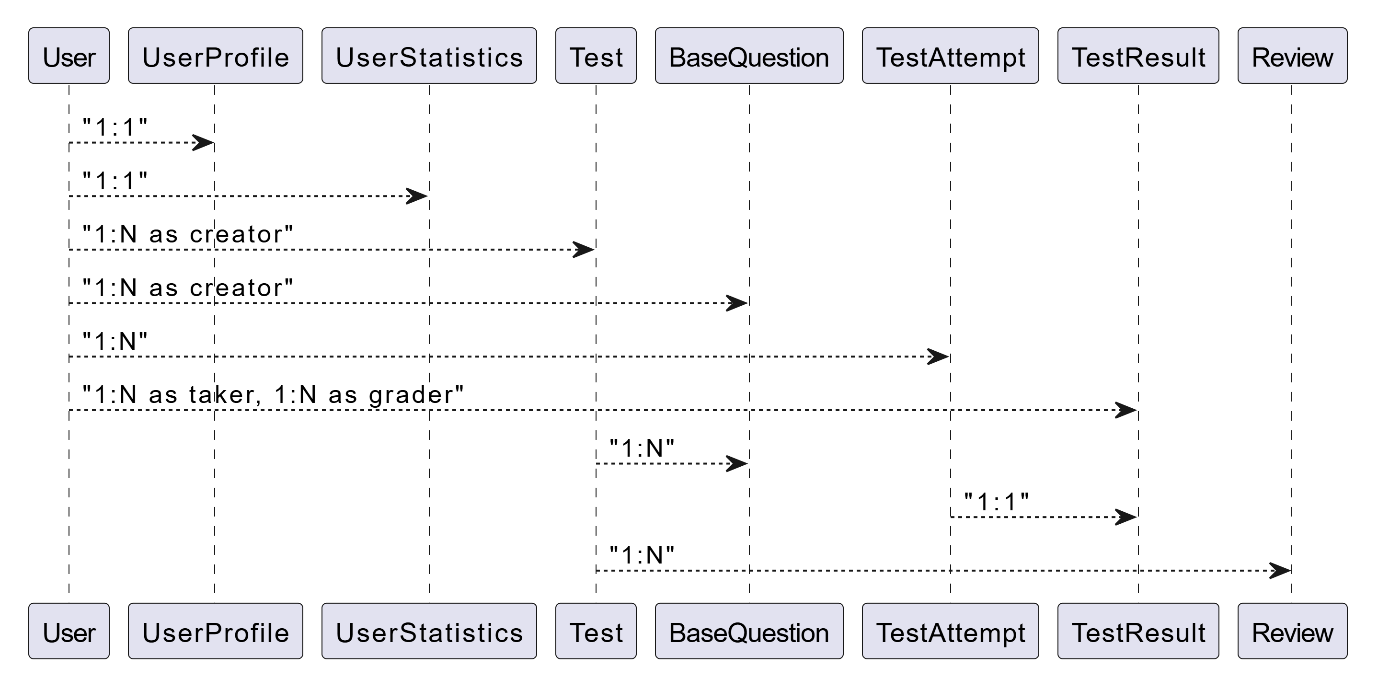
1. *BasePassage*

* *Class cơ sở cho các đoạn văn*
* *ReadingPassage: Đoạn đọc*
* *ListeningPassage: Đoạn nghe*

1. *Review*

* *Quản lý đánh giá và feedback cho bài test*

***1.2.2 Mối quan hệ giữa các classes***

******

1. *One-to-One Relationships*

* *User ←→ UserProfile*
* *User ←→ UserStatistics*
* *TestAttempt ←→ TestResult*

1. *One-to-Many Relationships*

* *User → Test (as creator)*
* *User → Question (as creator)*
* *User → TestAttempt*
* *User → TestResult (as taker)*
* *User → TestResult (as grader)*
* *Test → Question*
* *Test → Review*

1. *Inheritance Relationships*

* *BaseQuestion → Multiple Question Types*
* *BasePassage → Reading/Listening Passage*

*1.2.3. Đặc Điểm Chính*

*1. Modular Design*

* *Thiết kế module hóa cho phép dễ dàng mở rộng*
* *Tách biệt rõ ràng giữa các thành phần*

*2. Extensible Question System*

* *Sử dụng inheritance để tạo nhiều loại câu hỏi*
* *Dễ dàng thêm loại câu hỏi mới*

*3. Comprehensive Testing Framework*

* *Hỗ trợ nhiều loại bài kiểm tra*
* *Theo dõi chi tiết quá trình làm bài*
* *Phân tích kết quả đa chiều*

*4. User-Centric Architecture*

* *Tập trung vào trải nghiệm người dùng*
* *Theo dõi tiến trình học tập*
* *Cung cấp thống kê chi tiết*

*5. Flexible Assessment System*

* *Hỗ trợ nhiều loại đánh giá*
* *Tự động chấm điểm*
* *Feedback chi tiết*
  1. *Mô tả chi tiết*
     1. *Class User*

*Class User là trung tâm của hệ thống, quản lý thông tin người dùng và xác thực. Nó lưu trữ thông tin cơ bản và kết nối với các thông tin mở rộng thông qua UserProfile và UserStatistics.*

1. *Thuộc tính*
   * + 1. *Thông tin cơ bản*

* *email: String (: 8) - Mật khẩu đã mã hóa*
  + - 1. *Thông tin cá nhân*
* *firstName: String - Tên*
* *lastName: String - Họ*
* *dateOfBirth: Date - Ngày sinh*
* *phoneNumber: String - Số điện thoại*
* *address: Object {street, city, country} – Địa chỉ*
  + - 1. *Thông tin hệ thống*
* *avatar: String - URL ảnh đại diện*
* *isVerified: Boolean - Trạng thái xác thực*
* *isDeleted: Boolean - Trạng thái xóa*
* *role: String (enum: ["user", "teacher", "admin"]) - Vai trò*
* *userProfile: ObjectId - Liên kết với profile*
* *userStatistics: ObjectId - Liên kết với thống kê*

1. *Phương thức (methods)*
2. *Xác thực:*

* *comparePassword(candidatePassword: string): Promise<boolean>*
* *loginUser(email: string, password: string): Promise<User>*
* *logoutUser(id: string): Promise<User>*
* *generateToken(user: User): Promise<{accessToken, refreshToken}>*

1. *Quản lý người dung*

* *getUserById(id: string, select?: string, clean?: boolean): Promise<User>*
* *getProfileByUserId(id: string, select?: string, clean?: boolean): Promise<UserProfile>*
* *getFullUser(id: string, userOptions?: string, profileOptions?: string, clean?: boolean): Promise<User>*

1. *Cập nhật thông tin*

* *createUser(userData: object, profileData: object): Promise<{user, userProfile}>*
* *updateUser(id: string, updateData: object, profileUpdateData: object): Promise<{user, userProfile}>*
* *deleteUser(id: string): Promise<User>*
* *verifyUserEmail(id: string): Promise<User>*
* *changeUserPassword(id: string, newPassword: string): Promise<User>*
  + 1. *Class UserProfile*

*a. Thông tin cơ bản*

* *user: ObjectId (User)*
* *testTaken: Array<ObjectId> (Test)*
* *testScore: Array<Number>*
* *achievements: Array<String>*
* *preferences: {notificationSettings, studyReminders}*
* *learningGoals: Array<String>*
* *targetExams: Array<String>*
* *studyPlan: ObjectId (StudyPlan)*
* *progressTracking: {vocabularyMastered, grammarPointsLearned, listeningHoursPracticed, speakingSessionsCompleted, writingAssignmentsSubmitted}*
* *deviceInfo: Array<{deviceType, lastUsed}>*
* *bio: String*
* *website: String*
* *socialLinks: {twitter, linkedin, github}*
* *socialConnections: {studyBuddies, following, followers}*
* *feedbackProvided: Array<ObjectId> (Feedback)*

*b. Phương thức cơ bản*

* *addTestScore(score: number): Promise<UserProfile>*
* *updateProgressTracking(field: string, value: number): Promise<UserProfile>*
* *getUserProfile(userId: string, select?: string, clean?: boolean): Promise<UserProfile>*

*c. Thông tin hệ thống*

* + 1. *Class UserStatistic*

*a. Thông tin cơ bản*

* *user: ObjectId (User)*
* *testsCompleted: {total, byDifficulty: {easy, medium, hard},byType: {practice, exam}}*
* *averageScores: {overall, bySection: {reading, listening, writing, speaking}}*
* *questionStats: {totalAttempted, correctAnswers, byType: {single\_choice, multiple\_choice, fill\_in\_the\_blank}}*
* *timeStats: {totalTimeSpent, averageTestDuration,averageQuestionTime}*
* *progressOverTime: Array<{date, score, testId}>*
  + 1. *Class Test*

*Quản lý cấu trúc và nội dung của một bài kiểm tra, bao gồm các sections, câu hỏi, và các quy tắc chấm điểm*

*a. Thông tin cơ bản (thuộc tính)*

* *title: String (required, maxlength: 100)*
* *description: String (maxlength: 500)*
* *duration: Number (required, min: 1)*
* *totalPossibleScore: Number*
* *passingScore: Number*
* *difficulty: String*
* *testType: String*
* *createdBy: ObjectId (User)*

*b. Cấu trúc bài thi*

* *sections: Array<{*

*name: String (enum: ["reading", "listening", "speaking", "writing", "general"])*

*instructions: String*

*sectionScore: Number*

*duration: Number*

*questions: Array<{\_id: ObjectId, points: Number}>*

*passages: Array<{\_id: ObjectId, points: Number}>}>*

3. *Điều kiện trạng thái*

* *isPublished: Boolean*
* *attemptsAllowed: Number (min: 1)*
* *availableFrom: Date*
* *availableUntil: Date*
* *averageRating: Number (0-5)*
* *totalReviews: Number*
* *participantCount: Number*

1. *Phương thức*

* *updateAverageRating(): Promise<Test>*
* *publish(): Promise<Test>*
* *unpublish(): Promise<Test>*
  + 1. *Class TestAttempt*

*Quản lý 1 session làm bài test của người dung, theo dõi tiến trình và câu trả lời*

*a. Thuộc tính cơ bản*

* *user: ObjectId (User, required)*
* *test: ObjectId (Test, required)*
* *startTime: Date (required)*
* *endTime: Date*
* *maxEndTime: Date (required)*
* *status: String (enum: ["in-progress", "completed", "abandoned"])*
* *result: ObjectId (TestResult)*
* *totalScore: Number (default: 0, min: 0)*
* *answers: Array<{question: ObjectId (Question), answer: Mixed}>*

*b. Phương thức*

* *start(testAttempt: TestAttempt, test: Test): Promise<TestAttempt>*
* *isTimeUp(testAttempt: TestAttempt): boolean*
* *complete(testAttempt: TestAttempt): Promise<TestAttempt>*
* *calculateResult(testAttempt: TestAttempt): Promise<TestResult>*
* *calculateScore(questionData: object, answer: object, point: number): Promise<number>*
* *sanitizeJsonString(jsonString: string): string*
  + 1. *Class TestResult*

*Lưu trữ và phân tích kết quả chi tiết của một lần làm bài test*

*a. Thuộc tính cơ bản*

* *testAttempt: ObjectId (TestAttempt, required, unique)*
* *user: ObjectId (User)*
* *test: ObjectId (Test)*
* *totalScore: Number (required, min: 0)*
* *maxTotalScore: Number (default: 100)*
* *scorePercentage: Number (0-100)*
* *totalQuestions: Number (min: 1)*

*b. Điểm chi tiết*

* *sectionScores: Array<{sectionType: String, score: Number, totalQuestions: Number }>*
* *questionScores: Array<{id: ObjectId, score: Number}>*

*c. Kết quả kỹ năng nói và viết*

* *writingQuestionResults: Array<{question: ObjectId, aspects: Array<{aspect: String, score: Number, feedback: String}>, summary\_feedback: String, totalScore: Number }>*

* *speakingResult: {fluency: Number, pronunciation: Number, vocabulary: Number*

*overallCommunication: Number, totalScore: Number, feedback: String}*

*d. Feedback và đánh giá*

* *feedback: String*
* *gradedBy: ObjectId (User)*
* *gradedAt: Date*

*e. Phương thức: addFeedback(feedback: string, graderId: ObjectId): Promise<TestResult>*

* + 1. *Class Question*

*a. Thuộc tính cơ bản*

* *section: String (enum: ["reading", "listening", "speaking", "writing", "general", "vocabulary", "grammar"])*
* *questionText: String*
* *instruction: String*
* *isDraft: Boolean*
* *isPublic: Boolean*
* *points: Number*
* *difficulty: Number*
* *createdBy: ObjectId (User)*

*b. Các loại câu hỏi con*

* *MultipleChoice Question*
  + *answers: Array<String> (2-6 options)*
  + *correctAnswers: Array<String>*
* *SingleChoice Question*
  + *answers: Array<String> (2-6 options)*
  + *correctAnswer: String*
* *TrueFalse Question*
  + *correctAnswer: Boolean*
  + *statement: String*
* *FillInTheBlank Question*
  + *text: String*
  + *blanks: Array<{index: Number, correctAnswer: String, options?: Array<String> }>*
* *Matching Question*
  + *leftColumn: Array<{id: String, text: String}>*
  + *rightColumn: Array<{id: String, text: String}>*
  + *correctPairs: Array<{left: String, right: String}>*
* *Ordering Question*
  + *items: Array<{id: String, text: String}>*
  + *correctOrder: Array<String>*
* *OpenEnded Question*
  + *prompt: String*
* *Essay Question*
  + *minWords: Number (100-1000)*
  + *maxWords: Number (100-1000)*
  + *rubric: String*

*c. Các phương thức chung*

* *getQuestionById(id: string, clean?: boolean): Promise<Question>*
* *getQuestionsByIds(ids: string[], clean?: boolean, options?: object): Promise<Question[]>*
* *getQuestionsBySection(section: string, clean?: boolean): Promise<Question[]>*

1. *Đặc điểm chung:*
2. *Tính mở rộng: dễ dàng them câu hỏi mới, kế thừa các thuộc tính và phương thức cơ bản*
3. *Tính linh hoạt: hỗ trợ nhiều loại câu hỏi khác nhau, phù hợp với nhiều loại bài kiểm tra*
4. *Validation: Kiểm tra tính hợp lệ của câu trả lời, giới hạn về số lượng và định dạng options*
5. *Metada: lưu trữ thông tin về độ khó, theo dõi người tạo và trạng thái*
   * 1. *Class Review*

*a. Thông tin cơ bản*

* *testAttempt: ObjectId (TestAttempt)*
* *user: ObjectId (User)*
* *test: ObjectId (Test)*
* *totalScore: Number*
* *maxTotalScore: Number*
* *scorePercentage: Number*
* *totalQuestions: Number*
* *sectionScores: Array<{sectionType, score, totalQuestions}>*
* *questionScores: Array<{\_id, score}>*
* *writingQuestionResults: Array<WritingQuestionResult>*
* *speakingResult:{fluency, pronunciation, vocabulary, overallCommunication, totalScore, feedback}*
* *feedback: String*
* *gradedBy: ObjectId (User)*
* *gradedAt: Date*

# A screenshot of a computer*Sơ đồ trạng thái*

## Các Trạng Thái Chính

1. Trạng Thái Idle (Trạng thái bắt đầu)

* Người dùng đã đăng nhập vào hệ thống
* Có thể xem danh sách các bài thi
* Chưa bắt đầu làm bài thi nào

1. Test Selected (Đã chọn bài thi)

* Người dùng đã chọn một bài thi cụ thể
* Hệ thống kiểm tra điều kiện:
* Tính hợp lệ của subscription
* Số lần làm bài còn lại
* Thời gian có hiệu lực của bài thi

1. In Progress (Đang làm bài)

Bao gồm 3 trạng thái con:

1. Section Active

* Đang làm một phần cụ thể của bài thi
* Hệ thống theo dõi thời gian
* Tự động lưu câu trả lời
* Hiển thị câu hỏi và nhận câu trả lời

1. Section Paused

* Tạm dừng làm bài
* Lưu trạng thái hiện tại
* Có thể tiếp tục sau
* Thời gian tạm dừng

1. Section Completed

* Đã hoàn thành một phần
* Có thể chuyển sang phần tiếp theo
* Lưu trữ câu trả lời cuối cùng

1. Review Mode

* Xem lại toàn bộ câu trả lời
* Có thể quay lại chỉnh sửa
* Kiểm tra câu chưa trả lời
* Xác nhận trước khi nộp bài

1. Submitted (Đã nộp bài)

* Bài thi đã được nộp
* Không thể chỉnh sửa
* Bắt đầu quá trình chấm điểm
* Lưu trữ thời gian nộp bài

1. Graded (Đã chấm điểm)

* Hoàn thành chấm điểm
* Tính toán điểm số cuối cùng
* Cập nhật thống kê
* Hiển thị kết quả

## Các Chuyển Tiếp (Transitions)

1. Idle → Test Selected

* Trigger: Người dùng chọn bài thi
* Điều kiện:
* Bài thi còn hiệu lực
* Người dùng đủ điều kiện làm bài

1. Test Selected → In Progress

* Trigger: Người dùng bắt đầu làm bài
* Actions:
* Khởi tạo TestAttempt
* Bắt đầu tính thời gian
* Tải nội dung bài thi

1. Trong trạng thái In Progress

* Section Active → Section Paused:
* Trigger: Người dùng tạm dừng/lưu
* Action: Lưu trạng thái hiện tại
* Section Paused → Section Active:
* Trigger: Người dùng tiếp tục
* Action: Khôi phục trạng thái
* Section Active → Section Completed:
* Trigger: Hoàn thành phần hiện tại
* Action: Lưu câu trả lời và chuyển phần

1. In Progress → Review

* Trigger: Hoàn thành tất cả các phần
* Actions:
* Tổng hợp câu trả lời
* Hiển thị tổng quan

1. Review → Submitted

* Trigger: Người dùng xác nhận nộp bài
* Actions:
* Lưu tất cả câu trả lời
* Khóa khả năng chỉnh sửa
* Bắt đầu quá trình chấm điểm

1. Submitted → Graded

* Trigger: Hoàn thành chấm điểm
* Actions:
* Tính điểm tổng
* Cập nhật thống kê
* Tạo báo cáo kết quả

## Xử Lý Đặc Biệt

1. Tự động lưu

* Định kỳ lưu câu trả lời
* Theo dõi trạng thái làm bài
* Đảm bảo không mất dữ liệu

1. Kiểm soát thời gian
2. Theo dõi thời gian còn lại
3. Cảnh báo khi gần hết giờ
4. Tự động nộp bài khi hết giờ
5. Xử lý lỗi

* Mất kết nối internet
* Lỗi hệ thống
* Phục hồi trạng thái

1. Chấm điểm

* Tự động với câu hỏi khách quan
* Chờ chấm điểm thủ công với câu chủ quan
* Tổng hợp điểm từ nhiều phần