Jason

{

"users": {

"userId123": {

"username": "max",

"email": "max@example.com",

"role": "learner",

"points": 120,

"level": 3,

"createdAt": "2025-10-01T10:30:00Z",

"mistakes": {

"questionId\_42": {

"lastWrongAt": "2025-10-14T09:00:00Z"

}

},

"notes": {

"questionId\_42": {

"content": "Need to review this concept again."

}

}

}

},

"courses": {

"courseId\_uml": {

"title": "UML Basics",

"description": "Introduction to UML diagrams",

"difficulty": 1,

"requiredLevel": 1,

"isPublished": true,

"units": {

"unitId\_1": {

"title": "Class Diagrams",

"description": "Learn UML class diagrams",

"questionCount": 10,

"timeLimit": 600,

"unlockLevel": 1,

"questions": {

"questionId\_1": {

"questionText": "What does UML stand for?",

"answers": ["Unified Modeling Language", "Universal Markup Logic"],

"correctAnswerIndex": 0,

"points": 10,

"difficulty": 1,

"randomOrder": true

}

}

}

}

}

},

"attempts": {

"attemptId\_abc": {

"userId": "userId123",

"courseId": "courseId\_uml",

"unitId": "unitId\_1",

"score": 80,

"correctCount": 8,

"questionCount": 10,

"startTime": "2025-10-14T09:30:00Z",

"endTime": "2025-10-14T09:45:00Z"

}

},

"feedback": {

"feedbackId\_01": {

"userId": "userId123",

"courseId": "courseId\_uml",

"unitId": "unitId\_1",

"message": "Please add more diagrams examples.",

"createdAt": "2025-10-14T10:00:00Z"

}

}

}

+----------------+

| User |

+----------------+

| username: String

| email: String

| role: String

| points: Int

| level: Int

| createdAt: Timestamp

+----------------+

+----------------+

| Course |

+----------------+

| title: String

| description: String

| difficulty: Int

| requiredLevel: Int

| isPublished: Boolean

+----------------+

Relationship:

User 1..\* → Attempt

Course 1..\* → Unit

Unit 1..\* → Question

User 1..\* → Feedbac

@startuml

title UML Class Diagram – UML Learning App (Firestore-based)

'==============================

' Klassen

'==============================

class User {

+username: String

+email: String

+role: String

+points: Int

+level: Int

+createdAt: Timestamp

}

class Course {

+title: String

+description: String

+difficulty: Int

+requiredLevel: Int

+isPublished: Boolean

}

class Unit {

+title: String

+description: String

+questionCount: Int

+timeLimit: Int

+unlockLevel: Int

}

class Question {

+questionText: String

+answers: List<String>

+correctAnswerIndex: Int

+points: Int

+difficulty: Int

+randomOrder: Boolean

}

class Attempt {

+userId: String

+courseId: String

+unitId: String

+score: Int

+correctCount: Int

+questionCount: Int

+startTime: Timestamp

+endTime: Timestamp

}

class Feedback {

+userId: String

+courseId: String

+unitId: String

+message: String

+createdAt: Timestamp

}

class Note {

+questionId: String

+content: String

+createdAt: Timestamp

}

class Mistake {

+questionId: String

+lastWrongAt: Timestamp

}

'==============================

' Beziehungen (mit UML-Typen)

'==============================

' Aggregation – Ein User hat viele Notizen und Fehler, aber diese können getrennt existieren.

User o-- "\*" Note : has >

User o-- "\*" Mistake : records >

' Assoziation – Ein User kann mehrere Attempts und Feedbacks haben.

User --> "\*" Attempt : "makes"

User --> "\*" Feedback : "writes"

' Komposition – Ein Course besteht aus Units, die ohne Kurs nicht existieren.

Course \*-- "\*" Unit : "contains"

' Komposition – Eine Unit besteht aus mehreren Questions.

Unit \*-- "\*" Question : "includes"

' Assoziation – Attempt referenziert Course und Unit.

Attempt --> Course : "refers to"

Attempt --> Unit : "refers to"

' Assoziation – Feedback bezieht sich auf Course und Unit.

Feedback --> Course : "relates to"

Feedback --> Unit : "relates to"

@endumlk

Une image contenant texte, diagramme, reçu, Police

Le contenu généré par l’IA peut être incorrect.

Une image contenant Graphique, Police, rouge, logo

Le contenu généré par l’IA peut être incorrect.