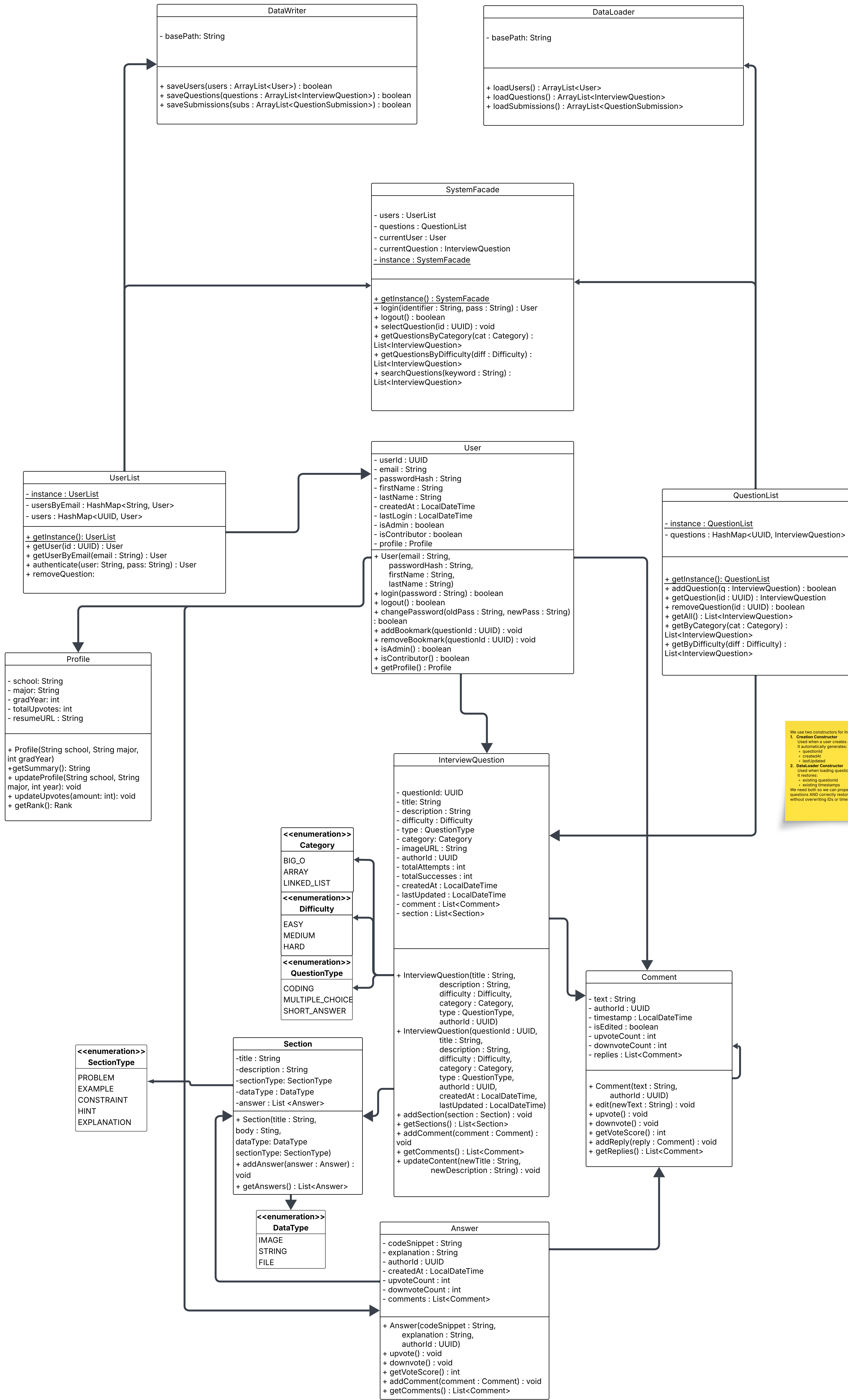


What is authorId?
authorId is NOT a new ID.
It stores the existing User.userId.
Example:
User.userId = 1111-AAAA
InterviewQuestion.authorId = 1111-AAAA
Answer.authorId = 1111-AAAA
Comment.authorId = 1111-AAAA
All of those point to the SAME user.
When we need the author:
User author = userList.getUser(authorId);
This follows the professor's rule:
"Reference IDs in JSON don't embed full objects."
So we store the ID, not the full User object.
Why Not Store author : User?
If we did:
author : User
Then questions.json would duplicate full user data inside every question/answer/comment.
That causes:
• Data duplication
• Heavy updates
• Bad JSON design
Using authorId avoids that.
Why Section, Answer, and Comment Don't Have UUIDs
They are nested objects.
• Section is inside InterviewQuestion
• Answer is inside Section
• Comment is inside Answer or Question
They are NOT stored independently.
So they don't need their own identity.



We use two constructors for InterviewQuestion:
1. **Creator Constructor**
Used when a user creates a new question.
If automatically generated:
• questionId
• createdAt
2. **Data-Loader Constructor**
Used when loading questions from JSON.
If restored:
• existing questionId
• existing timestamps
We need both so we can properly create new questions AND correctly restore saved data without overwriting IDs or timestamps.