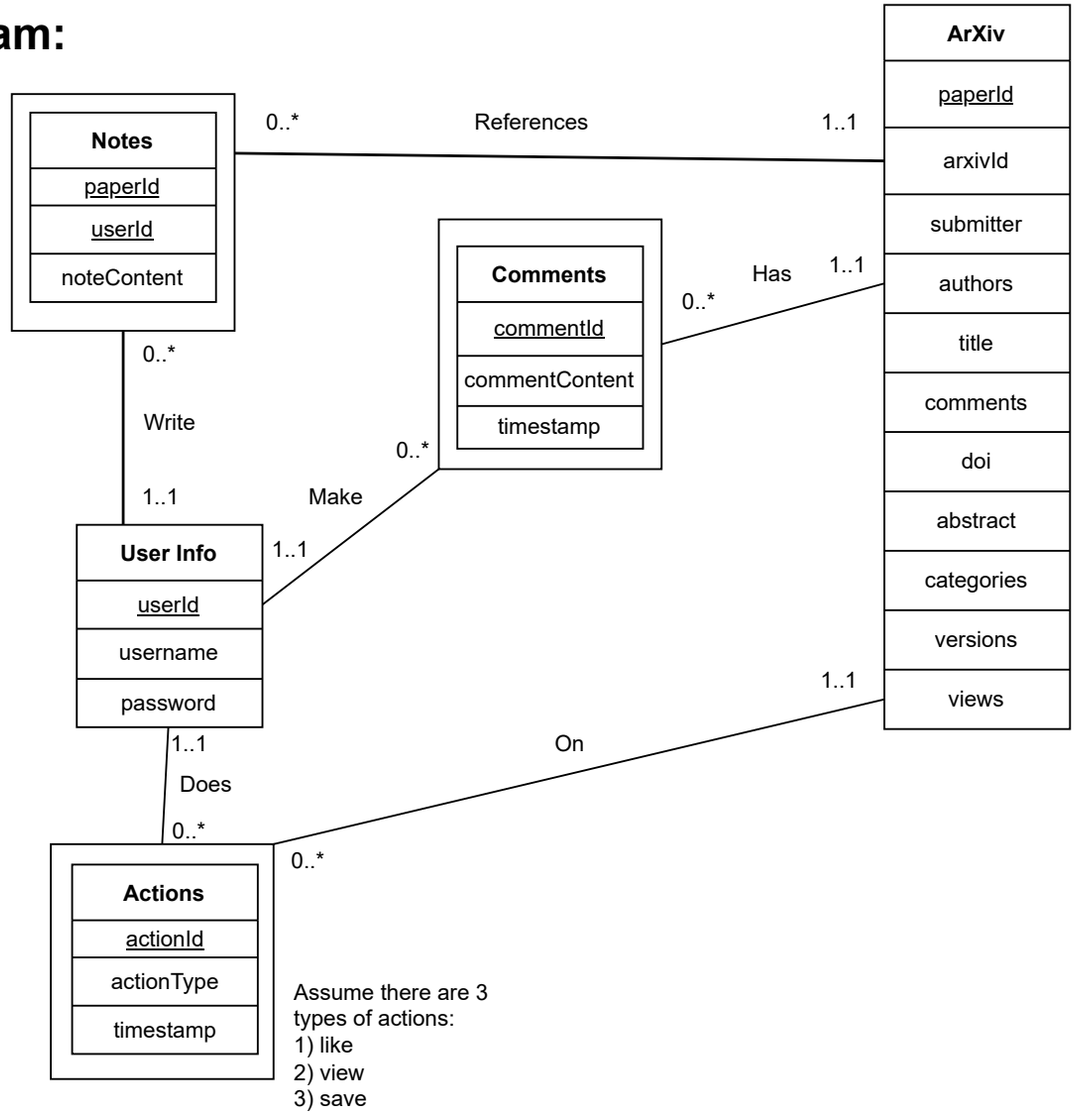
UML Diagram:



UML Descriptions:

User Info: The information a user needs to sign in. The userID is the Primary Key, so no two users can have the same ID.

ArXiv: The database we are using, which provides information for academic papers, and a section for how many times the paper has been viewed. The paperId is the Primary Key to distinguish between papers.

Notes: UserId, paperId -> Holds user-written content for the notes written for each paper. Uses UserID and PaperID as a combined Primary Key, so eachnote has 1 associated user and 1 associated paper.

Comments: commentId -> Holds user-written content for the notes written for each paper. CommentID as a Primary Key, and there can be many comments under 1 paper.

Actions: actionId -> actionType, Holds the actions that each user does (like, view, save). Is associated with a user and a paper as foreign keys, to say "X user commented on Y paper."

Relationship Cardinality Descriptions:

References: Each note references 1 particular ArXiv page. Each ArXiv page can be associated with many note pages. An ArXiv page can also not be referenced by any notes.

Write: The notes reference 1 particular user. 1 user can write many notes. A user can also write no notes.

Has: Each ArXiv page can have many comments on it. Comments only refer to 1 ArXiv page. It is possible for for an ArXiv page to have no comments.

Make: A user can make many comments, but each comment is associated with only 1 user. A user can have 0 comments.

On: An action only refers to 1 ArXiv Page. Many ArXiv pages can have an action done to them. An ArXiv page does not need to have any actions done on it.

Does: A user can do many actions, but each action is only done by one user. A user does not need to have any actions.

Functional Dependencies:

User Info: UserId -> username, password

Notes: UserId, paperId -> noteContent

Actions: actionId -> actionType, timestamp

ArXiv: paperId -> arxivId, submitter, authors, title, comments, doi, abstract, categories, versions

Comments: commentId -> commentContent, timestamp

BCNF/3NF Compliance:

We show our schema is BCNF compliant in our Minimal Basis. All left-hand sides are superkeys. Because it is BCNF, it is also 3NF.

Minimal Basis:

User Info: UserId -> Username, Password

ArXiv: paperId -> arxivId, submitter, title, comments, doi, abstract, categories, versions, views

Notes: UserId, paperId -> noteContent

Comments: commentId -> userId, paperId, commentContent, timestamp

Actions: actionId -> actionType, timestamp

Relational Schema:

User_Info(UserId:INT [PK], Username:VARCHAR(32), Password:VARCHAR(32))

ArXiv(paperId: INT [PK], arxivId: STRING, submitter: VARCHAR(100), authors: VARCHAR(8000), title: VARCHAR(500), comments: VARCHAR(1500), doi: VARCHAR (250), abstract: VARCHAR(8000), categories: VARCHAR(250), versions: INT, views: INT)

Notes(paperID: STRING [PK, FK], userID: INT [PK, FK], noteContent:VARCHAR(1000))

Comments(CommentID: INT [PK], UserId: INT [FK], paperId: STRING [FK], commentContent: VARCHAR(600), timestamp: TIMESTAMP)

Actions(actionId: INT [PK], userId: INT [FK], paperId: STRING [FK], savedAt: TIMESTAMP, actionType: VARCHAR(10))