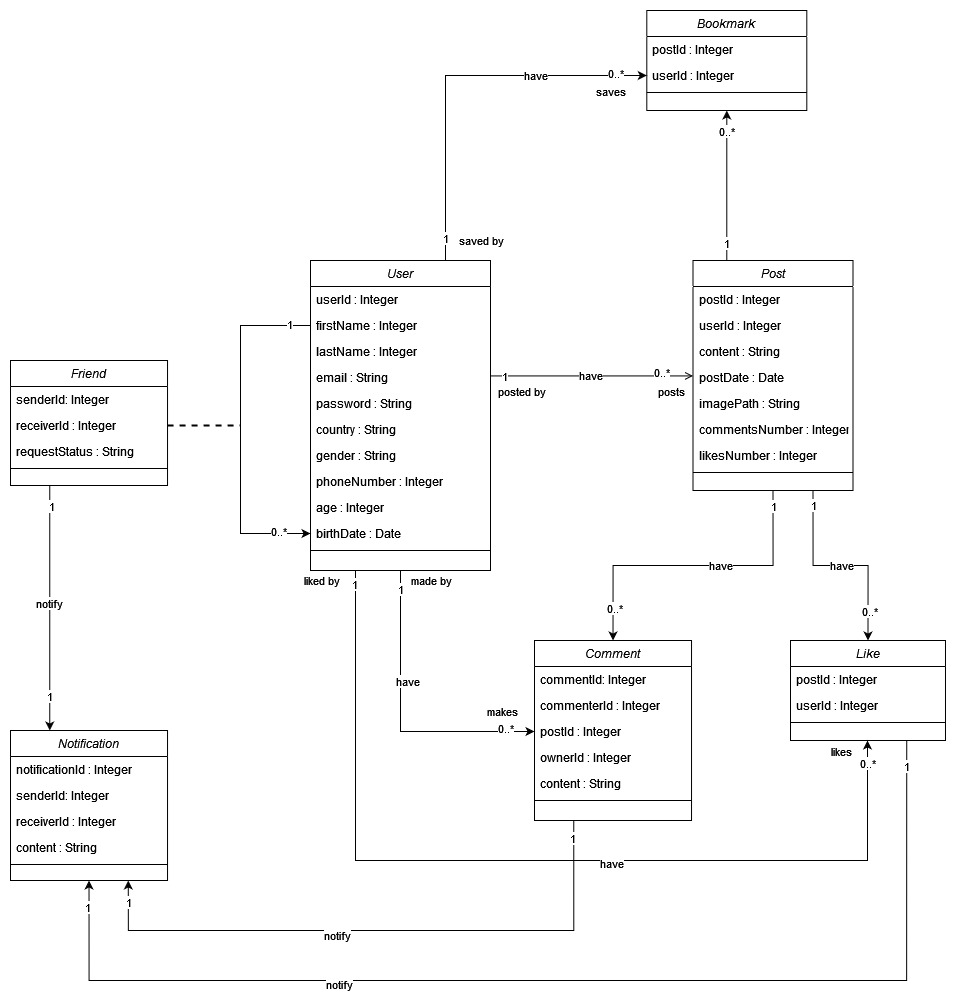
**OCL**

* **OCL associated with Class Diagram :**

****

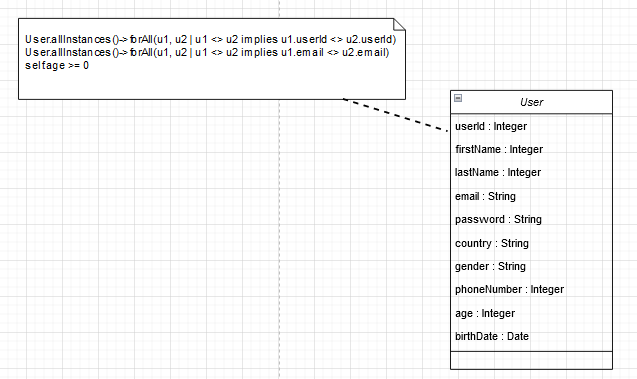
* **Constraints :**

**Module: User**

1. The **userId** attribute must be unique across all users.
2. The **email** attribute must be unique across all users.
3. The **age** attribute must be a non-negative integer.

**Invariant**

* User.allInstances() -> forAll(u1, u2 | u1 <> u2 implies u1.userId <> u2.userId)
* User.allInstances() -> forAll(u1, u2 | u1 <> u2 implies u1.email <> u2.email)
* self.age >= 0

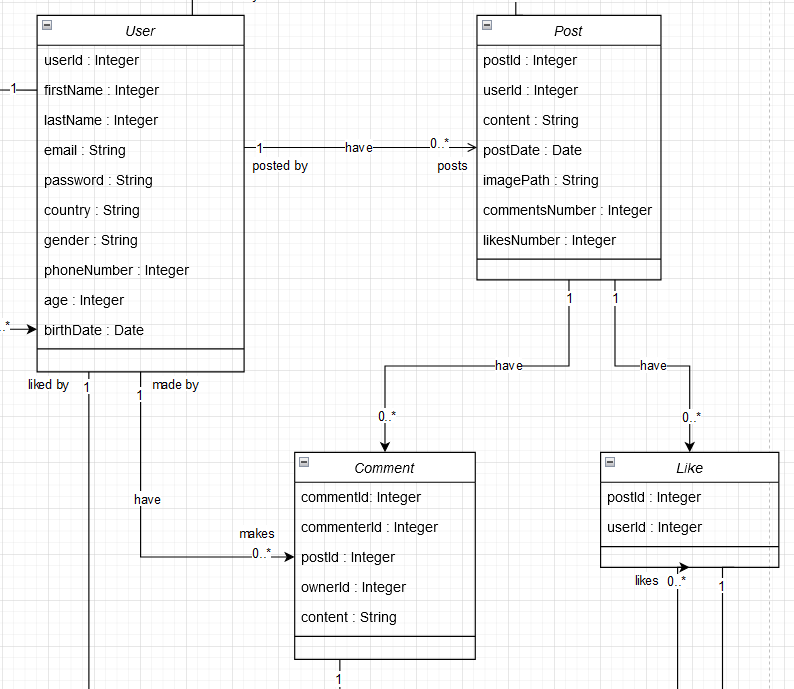


**Module: Post**

1. The **postId** attribute must be unique across all posts.
2. The **content** attribute must not be empty.
3. The **postDate** attribute must be a valid date.
4. Ensure that a **Post** is always associated with a valid **User**.
5. Ensure that commentsNumber in a **Post** is right.
6. Ensure that likesNumber in a **Post** is right.

**Invariant**

* Post.allInstances() -> forAll(p1, p2 | p1 <> p2 implies p1.postId <> p2.postId)
* not self.content.isEmpty()
* self.postDate <> null
* self.userId = posted\_by.userId
* posts.allInstances() -> forAll(p | p.commentsNumber = count(makes.postId = self.userId))
* posts.allInstances() -> forAll(p | p.likesNumber = count(likes.postId = self.userId))

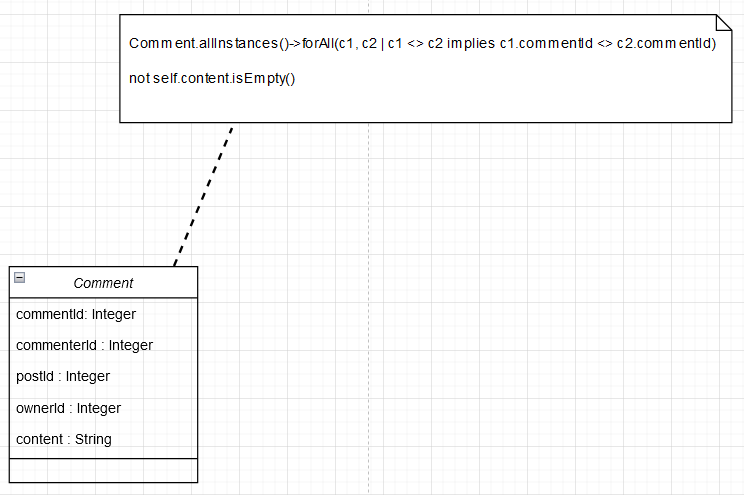
****

**Module: Comment**

1. The **commentId** attribute must be unique across all comments.
2. The **content** attribute must not be empty.

**Invariant**

* Comment.allInstances() -> forAll(c1, c2 | c1 <> c2 implies c1.commentId <> c2.commentId)
* not self.content.isEmpty()



**Module: Friends**

1. The **requestStatus** attribute must be one of the predefined values ("pending", "accepted").
2. Ensure that a friend request can only be sent to an existing user.

**Invariant**

* self.requestStatus = 'pending' or self.requestStatus = 'accepted'.
* User.allInstances() -> exists(u | u = self.sender or u = self.receiver)

**Module: Notifications**

1. The **notificationId** attribute must be unique across all notifications.

**Invariant**

* Notification.allInstances() -> forAll(n1, n2 | n1 <> n2 implies n1.notificationId <> n2.notificationId)