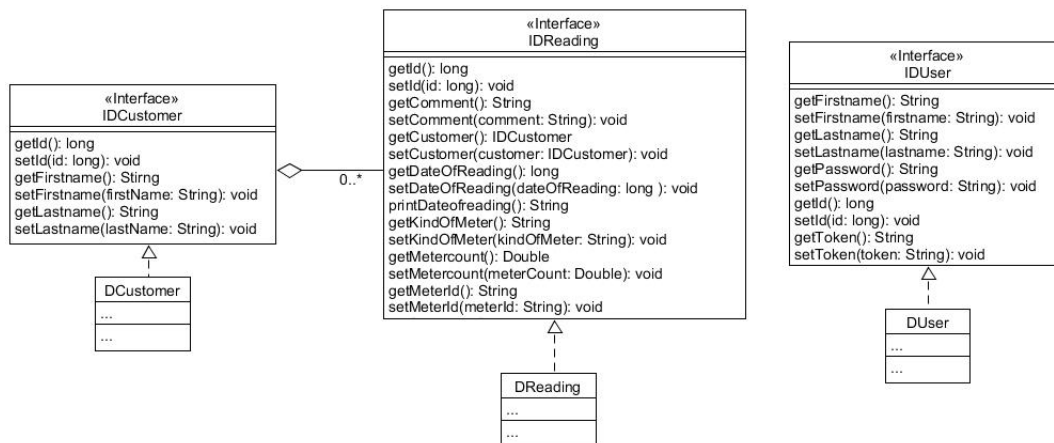


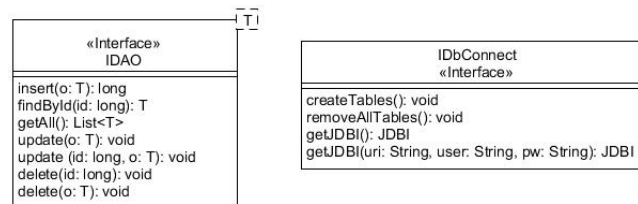
## Sprint 1: Datenbankanbindung der digitalen Hausverwaltung

**Abgabe:** Die vollständige Lösung zu dieser Aufgabe ist bis zum xx.xx.xxxx yy:yy Uhr in Ihr Repository einzupflegen.

In diesem Sprint geht es darum, für die Speicherung der Daten der digitalen Hausverwaltung eine entsprechende Datenbank bereitzustellen und mit Java entsprechende CRUD-Funktionalität umzusetzen. Die entsprechenden Datenhaltungsklassen für die Kunden-, Ablesungs- und Userdaten, die in der Datenbank gespeichert werden sollen, sollen sich dabei an folgendem Klassendiagramm orientieren:



Für die Verbindung zur Datenbank und den CRUD-Zugriff auf die Daten in der Datenbank sollen die Interfaces IDAO und IDbConnect die Grundlage sein:



Für den erfolgreichen Abschluss dieses Sprints sind folgende Punkte zu realisieren.

- Pflegen Sie die Maven-Projektvorlage in Ihr Git Repository ein.
- Erstellen Sie eine entsprechende Datenbank zur Speicherung der Kunden-, Ablesungs- bzw Userdaten. Dabei ist folgendes DBMS zu verwenden:  
SQLite
- Realisieren Sie anschließend mithilfe der Maven-Projektvorlage, unter Berücksichtigung der vorgegebenen Klassendiagramme, den CRUD-Zugriff auf die Daten in der Datenbank. Verwenden Sie dabei für den Zugriff auf die Datenbank das Framework JDBI.

Hinweis: Wird ein Kunde gelöscht, sollen dessen Ablesungen trotzdem gespeichert bleiben. Die Kundenreferenz soll bei den entsprechenden Ablesungen auf null gesetzt werden.

- Achten Sie darauf, dass Ihre Anwendung zu keiner Zeit durch Exceptions abstürzt. Formulieren Sie zudem aussagekräftige Fehlermeldungen.
- Erstellen Sie für Ihre Anwendung passende JUnit-Tests, so dass Sie für dieses Projekt eine Zweigabdeckung von mind. 80% erreichen.  
Hinweis: Die JUnit-Tests selbst sollen für die Zweigabdeckung nicht berücksichtigt werden. Automatisch generierter Code von Lombok dürfen für diese Zweigabdeckung ignoriert werden.

Jedes Teammitglied muss dabei zusätzlich die Skripte zu den folgenden Themen bearbeitet haben:

- JDBI
- Maven-Projekte
- Arbeiten mit git
- Testen mit JUnit