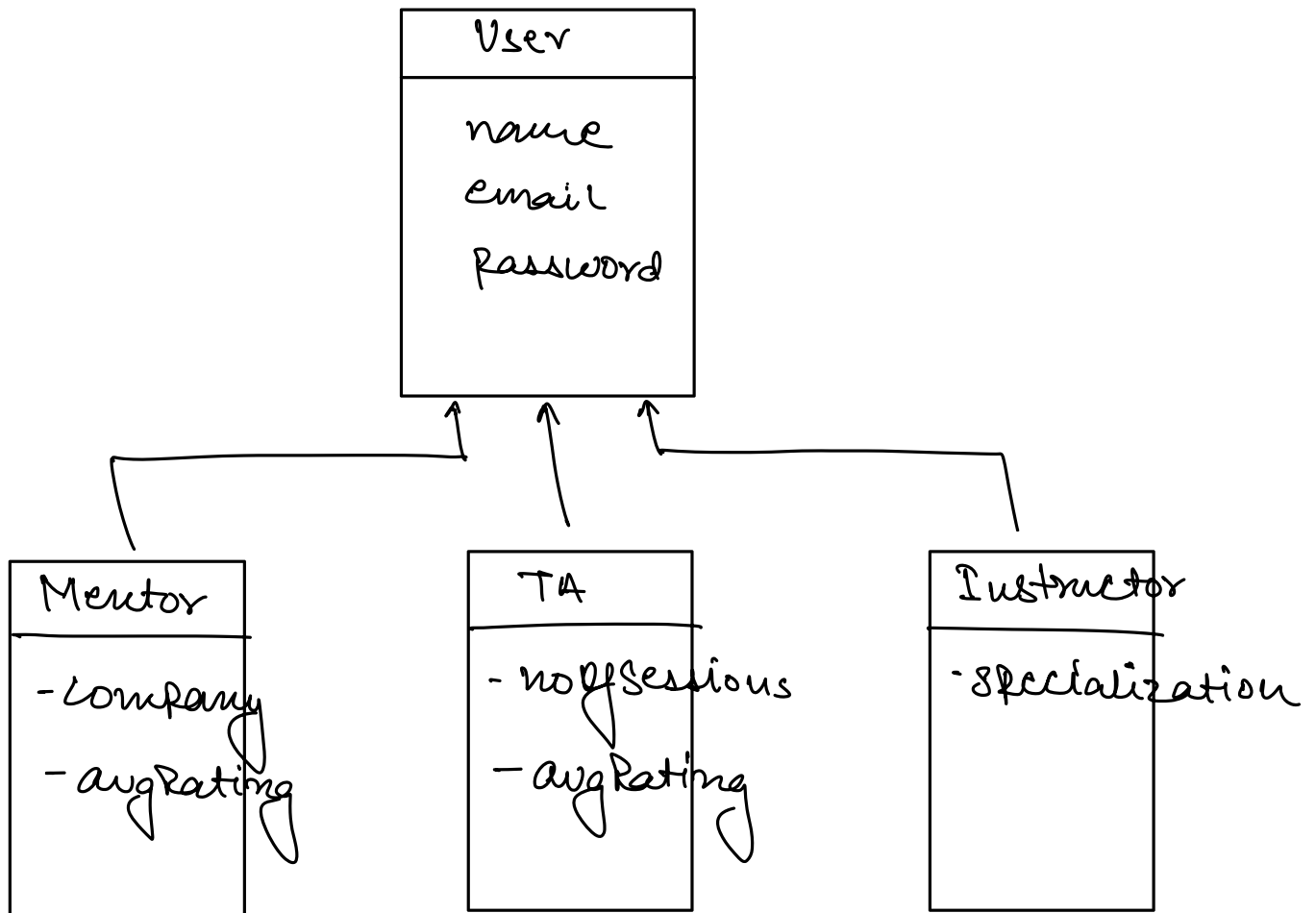


Agenda

- Representing Inheritance in DB.
- DB integration with our Product Service
- Demo for Representing Inheritance in DB.
- SelfProductService Implementation.



① MappedSuperClass.

⇒ When there's No object of parent class.

⇒ Parent class can be marked as an Abstract Class.

② Joined Table (Most frequently used way)

→ Every data wot objects of parent class, we'll store in the parent table.

→ for each ^{sub} class also, we'll create a table with only their own attributes.

→ We'll get parent class attrs in child classes via foreign key.

③ Table Per Class.

→ Exactly similar to MappedSuperClass, only difference, here we'll also create table for parent class as well.

→ table for each class will have their own attributes as well as parent class attributes.

④ Single Table.

⇒ Create one table with all the columns across the tables.

⇒ Add one extra column user-type to recognize the type of user.

Sparse Table ⇒ Table with lot of NULL values.

Database setup for ProductService

⇒ MySQL. ⇒ ORM. (Hibernate)

