

# Phase 2: UML and Front End Rapid Prototyping

## Notes:

1. A **User** can have multiple **Frontend** sessions, thus they are **Associated** with each other.
2. A **User** should be an **abstract** class with subclasses such as **Admin**, **BuyStandard**, **SellStandard**, **FullStandard**, which **inherit** from User, meaning there should be 4 **Inheritance** arrows pointing to User from Admin, BuyStandard, SellStandard, FullStandard.
3. A **User** can use multiple **Transactions**, thus they are **Associated** with each other.
4. A **Transaction** should be an **abstract** class with subclasses such as **Login**, **Logout**, **Create**, **Delete**, **Advertise**, **Bid**, **Refund**, **AddCredit**, **ViewAuctions**, **ViewUsers**, which inherit from Transaction, meaning there should be 10 **Inheritance** arrows pointing to Transaction from all the transactions.
5. An **Item** should **Depend** on a **User** as a User can own an Item.
6. **Advertise**, **Bid** should **Depend** on **Item** as they either create or edit an Item

**Transaction:** An abstract class that initializes processing transactions.

**Login, Logout, Create, Delete, Refund, AddCredit, ViewUsers, Bid, Advertise,**

**ViewAuctions:** Transaction subclasses that inherit from Transaction, override processTransaction(), and handle each transaction appropriately.

**User:** An abstract class that initializes user types.

**Admin, FullStandard, BuyStandard, SellStandard:** User subclasses that inherit from User.

**Frontend:** The main frontend session that handles relationships between Items, Users, and Transactions.

**Item:** A user-owned item that relates to bidding, advertising, and viewing.