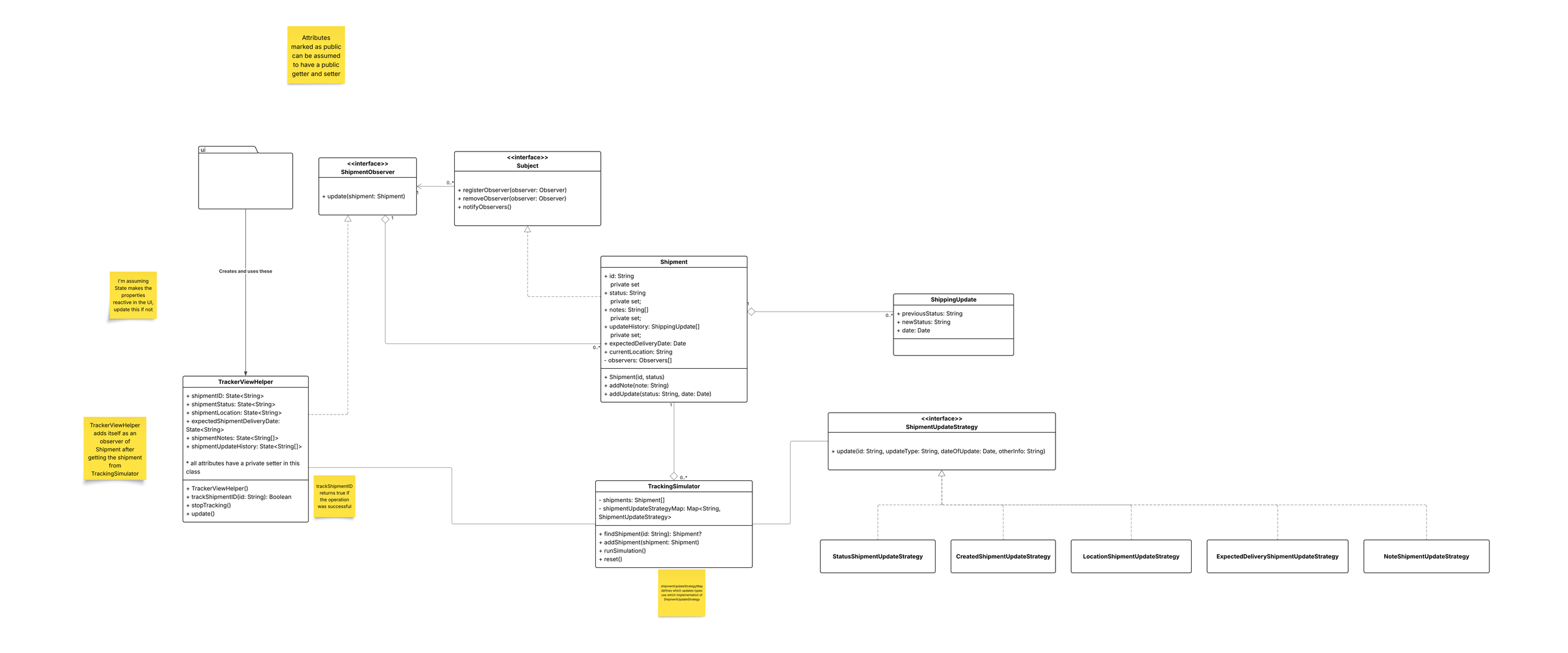
Attributes marked as public can be assumed to have a public getter and setter
ui
Creates and uses these
I'm assuming State makes the properties reactive in the UI, update this if not
<<interface>>
ShipmentObserver
+ update(shipment: Shipment)
<<interface>>
Subject
+ registerObserver(observer: Observer)
+ removeObserver(observer: Observer)
+ notifyObservers()
0..*
1
1
Shipment
+ id: String
private set
+ status: String
private set;
+ notes: String[]
private set;
+ updateHistory: ShippingUpdate[]
private set;
+ expectedDeliveryDate: Date
+ currentLocation: String
- observers: Observers[]
+ Shipment(id, status)
+ addNote(note: String)
+ addUpdate(status: String, date: Date)
1
0..*
ShippingUpdate
+ previousStatus: String
+ newStatus: String
+ date: Date
1
0..*
TrackerViewHelper
+ shipmentID: State<String>
+ shipmentStatus: State<String>
+ shipmentLocation: State<String>
+ expectedShipmentDeliveryDate: State<String>
+ shipmentNotes: State<String[]>
+ shipmentUpdateHistory: State<String[]>
* all attributes have a private setter in this class
+ TrackerViewHelper()
+ trackShipmentID(id: String): Boolean
+ stopTracking()
+ update()
TrackerViewHelper adds itself as an observer of Shipment after getting the shipment from TrackingSimulator
trackShipmentID returns true if the operation was successful
TrackingSimulator
- shipments: Shipment[]
- shipmentUpdateStrategyMap: Map<String, ShipmentUpdateStrategy>
+ findShipment(id: String): Shipment?
+ addShipment(shipment: Shipment)
+ runSimulation()
+ reset()
0..*
shipmentUpdateStrategyMap defines which updates types use which implementation of ShipmentUpdateStrategy
<<interface>>
ShipmentUpdateStrategy
+ update(id: String, updateType: String, dateOfUpdate: Date, otherInfo: String)
StatusShipmentUpdateStrategy
CreatedShipmentUpdateStrategy
LocationShipmentUpdateStrategy
ExpectedDeliveryShipmentUpdateStrategy
NoteShipmentUpdateStrategy