public class AgentControl {

private AgentBorder agentBorder;

private AgentEntity agentEntity;

private CustomerEntity customerEntity;

private ArrayList<ArrayList<ItineraryCaseEntity>> itineraries;

private ArrayList<PriceWatchEntity> watches;

//Accessors and Mutators//

public AgentBorder getAgentBorder() {

return agentBorder;

}

public void setAgentBorder(AgentBorder agentBorder) {

this.agentBorder = agentBorder;

}

public AgentEntity getAgentEntity() {

return agentEntity;

}

public void setAgentEntity(AgentEntity agentEntity) {

this.agentEntity = agentEntity;

}

public CustomerEntity getCustomerEntity() {

return customerEntity;

}

public void setCustomerEntity(CustomerEntity customerEntity) {

this.customerEntity = customerEntity;

}

public ArrayList<ArrayList<ItineraryCaseEntity>> getItineraries() {

return itineraries;

}

public void setItineraries(ArrayList<ArrayList<ItineraryCaseEntity>> itineraries) {

this.itineraries = itineraries;

}

public ArrayList<PriceWatchEntity> getWatches() {

return watches;

}

public void setWatches(ArrayList<PriceWatchEntity> watches) {

this.watches = watches;

}

//Other Methods//

private void LogIn(){

//Have to do anything here?

}

private CustomerEntity CreateCustomerAccount(String name, String address, String emailAddress, String phoneNumber, CreditCardEntity creditCard){

return new CustomerEntity();

}

private CustomerEntity UpdateCustomer(int customerID, String name, String address, String emailAddress, String phoneNumber, CreditCardEntity creditCard){

//Return Updated Existing CustomerEntity!!!

return new CustomerEntity();

}

private ItineraryCaseEntity CreateNewItinerary(int customerID, String departureCity, String arrivalCity, GregorianCalendar depatureDate, GregorianCalendar returnDate, String Preference){

return new ItineraryCaseEntity();

}

private ArrayList<ArrayList<ItineraryCaseEntity>> SearchFlights(){

//Is this what you meant???

return this.getItineraries();

}

private ItineraryCaseEntity ReserveFlight(){

//Return New ItineraryCaseEntity?

return new ItineraryCaseEntity();

}

private ItineraryCaseEntity ModifyReservation(int customerID, String departureCity, String arrivalCity, GregorianCalendar departureDate, GregorianCalendar returnDate, String preference){

//Return Updated Existing ItineraryCaseEntity!!!

return new ItineraryCaseEntity();

}

private void CancelReservation(){

}

private PriceWatchEntity CreatePriceWatch(double priceDesired){

//TODO: Add More Info to parameters ^

return new PriceWatchEntity();

}

private void CancelPriceWatch(){

//This is based on when a watch expires? What about when a customer opts to cancel a watch?

}

private double computeCost(){

double cost = 0;

//Calculate Cost//

return cost;

}

private ArrayList<PriceWatchEntity> ProvideMetWatches(){

//TODO: Discuss this!!!

ArrayList<PriceWatchEntity> metWatches = null;

//Traverse watches, adding any "met" watches to the metWatches ArrayList.

return metWatches;

}

private Document ProduceDocument(String documentType){

//TODO: Add More Info

return new Document();

}

private void ProduceFlightReceipt(){

}

private void ProduceFlightList(){

}

}