Demo video link

(slides follow)
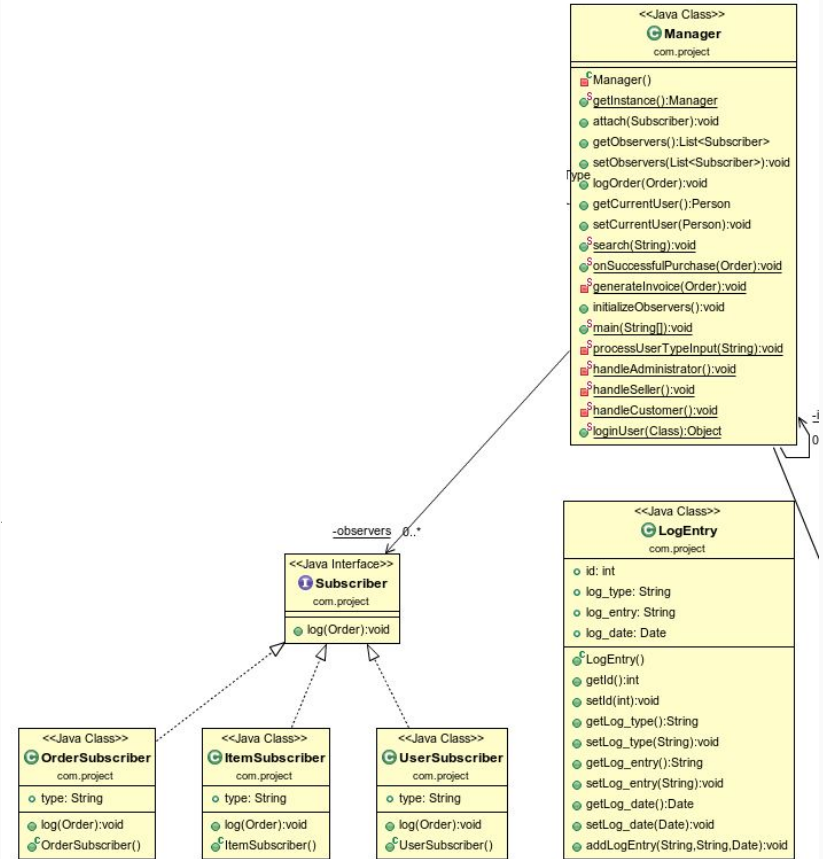
Marketplace (Group 31)

Rohit Gupta
Corin Sandford
Mike Gore

<<Java Class>>
Manager
com.project

- Manager()
- getInstance():Manager
- attach(Subscriber):void
- getObservers():List<Subscriber>
- setObservers(List<Subscriber>):void
- logOrder(Order):void
- getCurrentUser():Person
- setCurrentUser(Person):void
- search(String):void
- onSuccessfulPurchase(Order):void
- generateInvoice(Order):void
- initializeObservers():void
- main(String[]):void
- processUserTypeInput(String):void
- handleAdministrator():void
- handleSeller():void
- handleCustomer():void
- loginUser(Class):Object

-observers 0..*

<<Java Interface>>
Subscriber
com.project

- log(Order):void

<<Java Class>>
LogEntry
com.project

- id: int
- log_type: String
- log_entry: String
- log_date: Date

- LogEntry()
- getId():int
- setId(int):void
- getLog_type():String
- setLog_type(String):void
- getLog_entry():String
- setLog_entry(String):void
- getLog_date():Date
- setLog_date(Date):void
- addLogEntry(String,String,Date):void

<<Java Class>>
OrderSubscriber
com.project

- type: String

- log(Order):void
- OrderSubscriber()

<<Java Class>>
ItemSubscriber
com.project

- type: String

- log(Order):void
- ItemSubscriber()

<<Java Class>>
UserSubscriber
com.project

- type: String

- log(Order):void
- UserSubscriber()

```java
@Entity
@Table(name = "log",
        uniqueConstraints={@UniqueConstraint(columnNames={"log_id"})})
public class LogEntry {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    @Column(name = "log_id", nullable = false, unique = true)
    public int id;

    @Column(name = "log_type")
    public String log_type;

    @Column(name = "log_entry")
    public String log_entry;

    @Column(name = "date")
    public Date log_date;
```
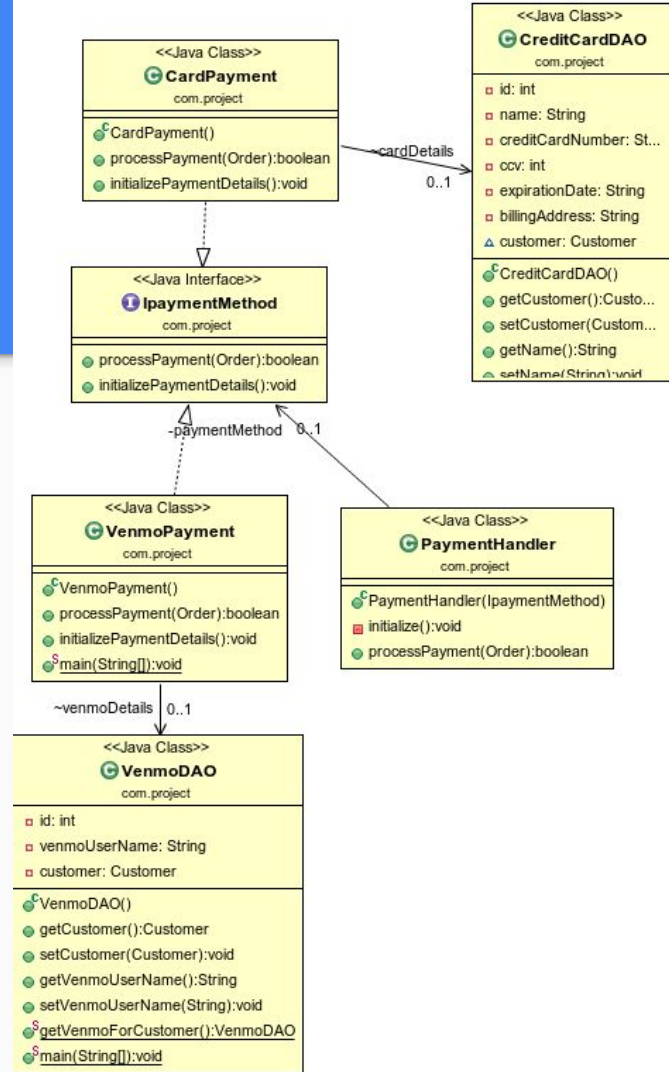
Design Pattern: Strategy

Used for Credit Card payments

```java
@Entity
@Table(name = "customer")
@AttributeOverride(name = "id", column = @Column(name = "customer_id"))
public class Customer extends Person {

    // TODO
    @Transient
    private Membership membership;

    @OneToOne(mappedBy = "customer", cascade = { CascadeType.ALL }, fetch = FetchType.LAZY)
    private Cart cart;

    @Transient
    private HashMap<Integer, Order> user_orders = new HashMap<>();
```

Strategy Pattern

```java
@Entity
@Table(name = "cart", uniqueConstraints = { @UniqueConstraint(columnNames = { "cart_id" }) })
public class Cart {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "cart_id", nullable = false, unique = true, length = 11)
    private int id;

    @ElementCollection
    @MapKeyJoinColumn(name = "item_id")
    @Column(name = "quantity")
    @CollectionTable(name = "cart_items", joinColumns = @JoinColumn(name = "cart_id"))
    private Map<Item, Integer> itemCountMap;

    @OneToOne(optional = false)
    @JoinColumn(name = "customer_id")
    private Customer customer;
```

Thanks!