

Class Diagram

- * Use Crud to start building from
- * Take note to toggle between training & Prod mode will be done via the configuration file
- * The configuration file will write to Orders_DDMMYYY.txt

The UI should start with a menu to prompt the user for what they would like to do:

```
* * * * *
*  <<Flooring Program>>
*  1. Display Orders
*  2. Add an Order
*  3. Edit an Order
*  4. Remove an Order
*  5. Save Current Work
*  6. Quit
*
* * * * *
* *
```

Example :

```
> package name
  +class name : interface/inheritance
    + public name : return type
    - private name : return type
```

```
*****
*****
```

* Ask Does Full Crud Apply to all of the DTO's?

```
> com.mr.FlooringOrder.dtos
```

```
+ Taxes
  - State : String
  - TaxRate : BigDecimal

+ Products
  - ProductType : String
  - CostPerSquareFoot : Double
  - LaborCostPerSquareFoot : Double

+ Orders
  - Order # : int
  - Customer Name : String
  - State : String
```

- Data/ Taxes : BigDecimal
- Product Type : String
- Area : Double
- Cost per Sq Foot : BigDecimal
- Labor Cost Per Sq Foot : BigDecimal
- Material Cost : BigDecimal
- Labor Cost : BigDecimal
- Tax : BigDecimal
- Total : BigDecimal

>com.mr.FlooringOrder.daos

- + OrderDao
 - + Create(Order order) : Order
 - + ReadAll() : List <Order>
 - + Update(Int order#) : Order
 - + Delete(Int order#) : Order
 - + Save (Int order #) : Order
- + OrderDaoFileProdImpl : OrderDao
 - orders : Map< <Order>
 - FILENAME : String
 - + OrderDaoFileImpl(String fileName)
 - save() : void throws PersistenceException
 - load() : void throws PersistenceException
- + OrderDaoFileTrainingImpl : OrderDao
 - orders : Map <Order>
 - FILENAME : String
 - + OrderDaoFileImpl(String fileName)
 - load() : void throws PersistenceException
- + StateTaxDao
 - + ReadAll() : HashMap <StateTax> StateTax
 - + Read() : StateTax
- + ProductDao
 - + ReadAll() : List <ProductDao>

>com.mr.FlooringOrders.Service

- +FlooringOrderService
 - Exception e : Exception
 - DuplicateOrderException : Exception
 - InvalidOrderException : Exception
 - OrderNotFoundException : Exception
 - PersistenceException : Exception
 - + DisplayAllOrders : Map <Order> throws Exception e

```

        + AddOrder(Order order): Order throws
DuplicateOrderException , InvalidOrderException
        + EditOrder(Order order) : Order throws
InvalidOrderException
        + RemoveOrder(Order order) : Order throws
InvalidOrderException, OrderNotFoundException
        + SaveOrder(Order order) : Order throws
PersistenceException

        + getCustomerName(String) : Order // ??? Do we make 4
separate methods for the view to be checked in the service Layer

```

```

        + FlooringOrderServiceImpl : FlooringOrderService
        - FlooringOrderDao : FlooringOrderDao
        - StateTaxDao : StateTaxDao
        - ProductDao : ProductDao
        + FlooringOrderServiceImpl(OrderDao orderDao,
ProductDao productDao)

```

```

> com.mr.FlooringOrders.ui
+FlooringOrders.view
    + getMenuSelection() : int
    + addOrderNumber() : int
    + getCustomersName() : String
    + getState() : String
    + getArea() : double
    + getProductType() : String
    + getStateTaxRate() : BigDecimal
    + displayAllOrders(List<Order> orders) : void

```

```

> com.mr.FlooringOrders.controller
+ FlooringOrdersController
    - service : FlooringOrdersService
    - view : FlooringOrdersView
    + FlooringOrdersController(FlooringOrdersService service,
FlooringOrdersView view)
    + run() : void
    - addOrder() : void
    - editOrder() : void
    - removeOrder() : void
    - saveUser() : void
    - viewAllOrders() : void

```