

```
1
2
3  Aramco Trading {
4
5      [IS442_G1T6 Presentation]
6
7
8
9      < Energy Trading Data Application >
10
11
12  }
13
14
```

Table Of 'Contents' {

01 Client-Server Architecture
< Components and Interactions >

02 Application Layer
< Structure for the application >

03 OOP Design Principles
< Doing it the right way >

04 Open-source Libraries
< Options and Considerations >

05 Demo
< THE REAL DEAL >

}

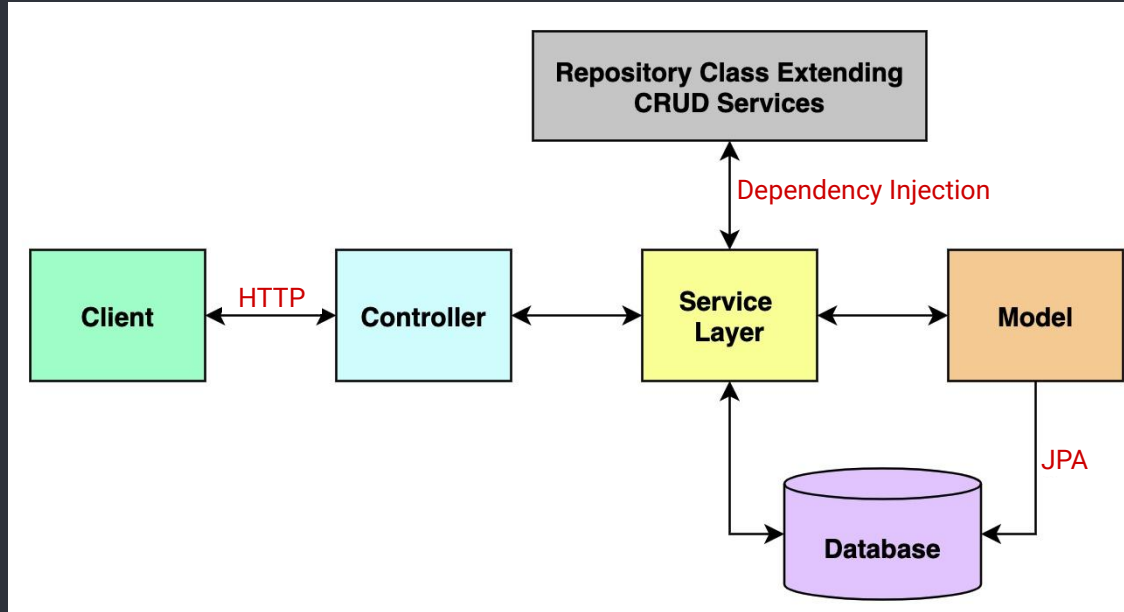
01 {

[Client-Server
Architecture]

< Components and Interactions >

}

Components + Interactions; {



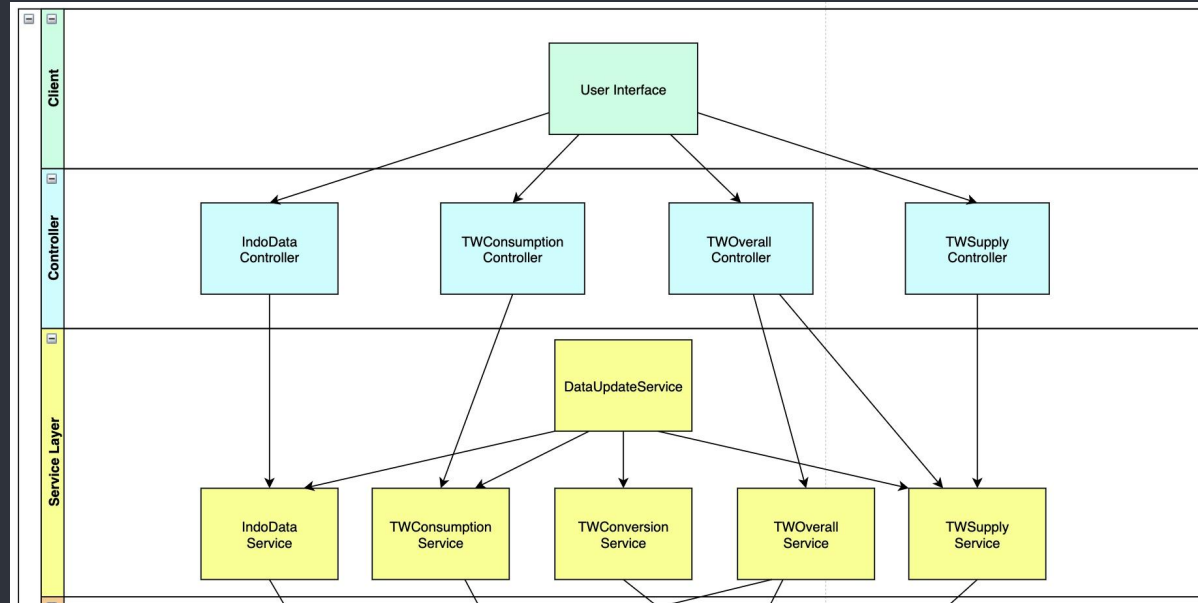
02 {

[Application Layers]

< Structure for the application >

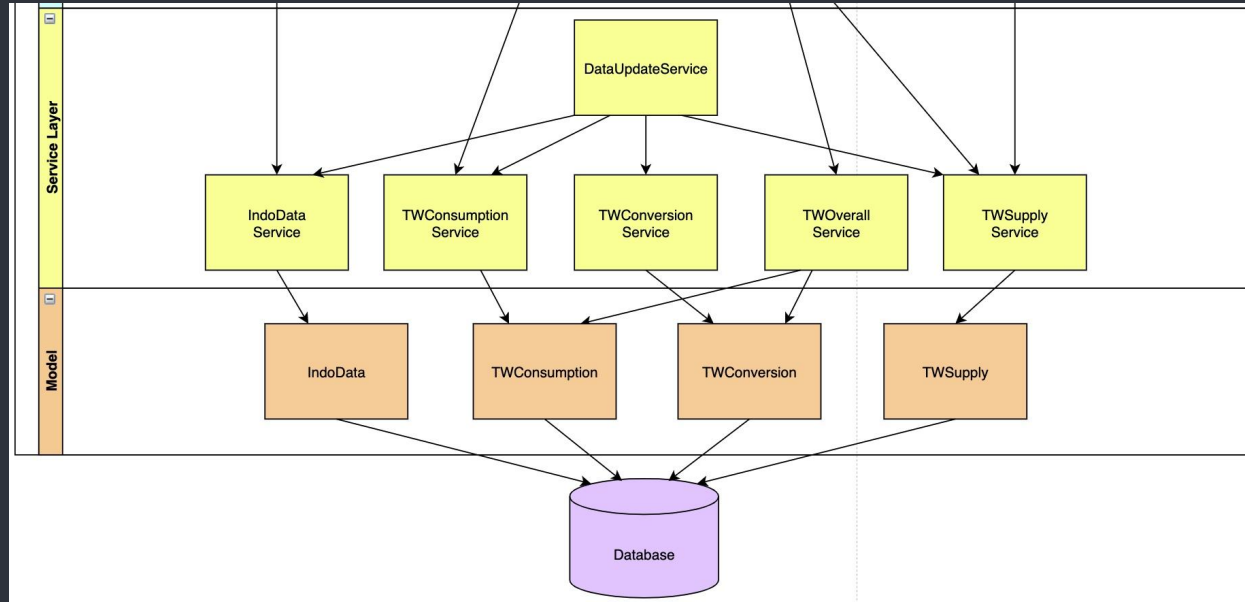
}

Layers_1; {



}

Layers_2; {



}

03 {

[OOP Design Principles]

< Doing it the right way >

}


```
1 Principles_1; {
```



```
3 Encapsulation ✓
```

```
4 < Appropriate states for class attributes and methods >
```



```
6
```

```
7 Abstraction ✓
```

```
8 < Each class only contains the relevant methods >
```



```
10
```

```
11 Inheritance ✓
```

```
12 < Acquire attributes and methods of parent class >
```

```
13
```

```
14 }
```

```
1 Principles_2; {
```

```
2
```

```
3
```

```
4
```

```
5
```



Single Responsibility Principle 

```
6
```

< Each component only has one responsibility >

```
7
```

```
8
```

```
9
```



Interface Segregation Principle 

```
10
```

< Each service class implements its own interface class >

```
11
```

```
12
```

```
13
```

```
14
```

```
}
```

1
2 04 {
3
4

5 [Open-source Libraries]
6
7

8 < Options and Considerations >
9
10

11 }
12
13
14

Front-end; {



ApexCharts.js

< New, Neat, Responsive, Interactive,
BUT Laggy with larger datasets >



D3.js

< Capabilities beyond data visualisation,
BUT Steep learning curve, Outdated documentation >





Chart.js



< Lightweight, Simple, Flexible, Responsive, Interactive >

}

```
1 Front-end; {
```

```
2  
3 [  ] Bootstrap   
4 < CSS, JS >
```

```
5  
6  
7 [  ] Iconify   
8 < Indonesia & Taiwan flag icons >
```

```
9  
10  
11 [  ] Font Awesome   
12 < Home icon >
```

```
13 }  
14
```

Back-end; {



Struts

< Framework based on the MVC architecture
BUT Heavyweight framework,
Tightly coupled programming modules >



Spring Boot

< Extension of Spring framework to develop a
Spring web application >

}

Back-end; {



MongoDB

< NoSQL Database

BUT meant for semi/non-structured data >



MySQL 

< Relational Database Management System >

}

Back-end; {



JSoup

< Extract data from websites, Simple to use,
BUT Unable to perform web crawling >



Selenium



< Web crawling, Able to interact with websites >

}

Back-end; {



Groupdocs.conversion



< Convert files scraped to .xlsx >



Lombok



< Minimize or remove boilerplate codes. Saves space and enhances readability of source code >



Apache.POI



< Used to design or modify Microsoft Office files with java program >

}

1
2
3
4
5
6
7
8
9
10
11
12
13
14

05 {

[Demo]

< THE REAL DEAL >

}

```
1 Thank_you; {  
2  
3  
4  
5  
6     System.out.println(  
7         "Do you have any questions?"  
8     );  
9  
10  
11  
12  
13  
14 }
```