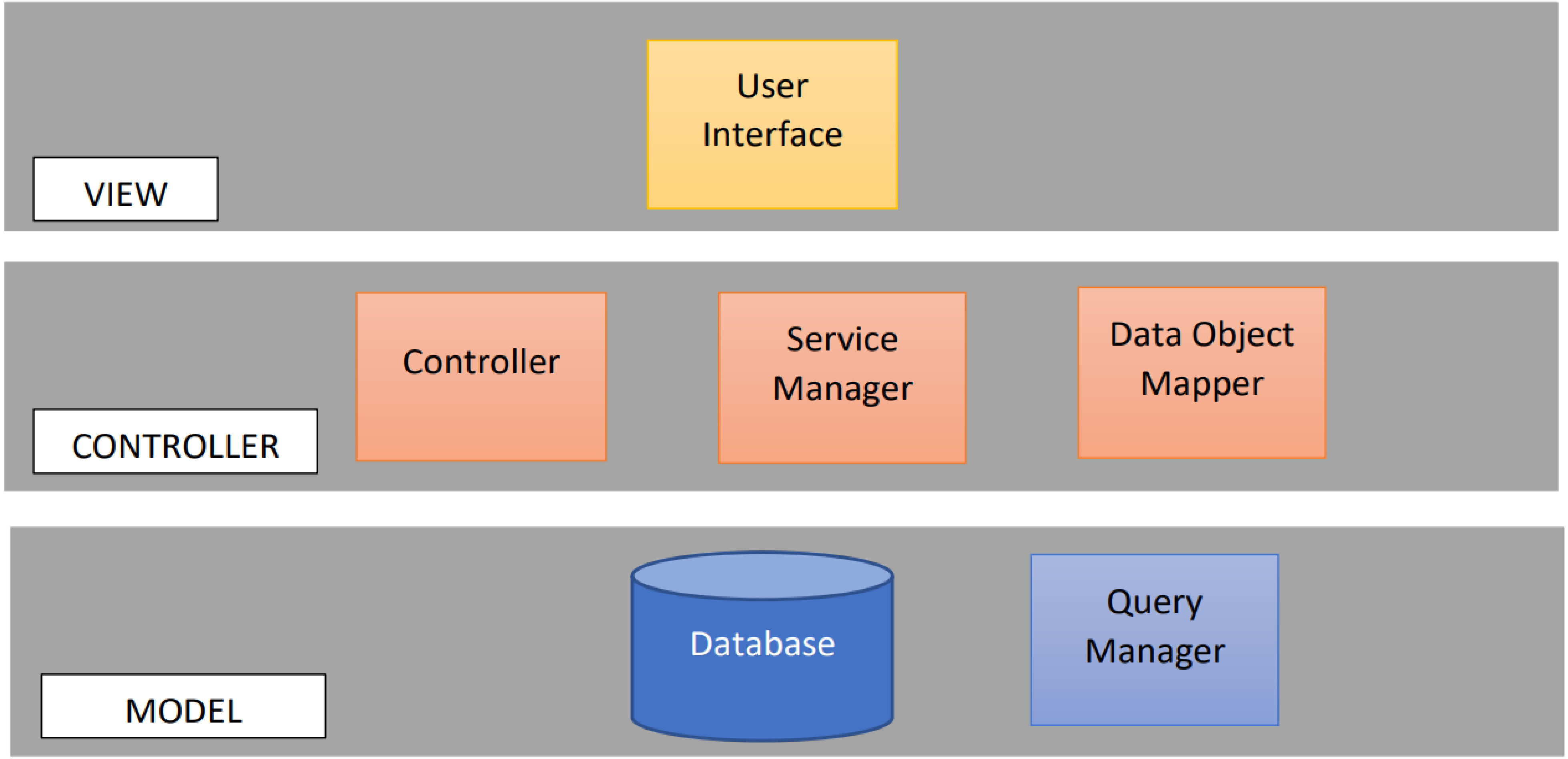
APP Project Demo

SOURAV GANESHA | SINDOORA S RAO

Architectural Overview



VIEW

User Interface

CONTROLLER

Controller

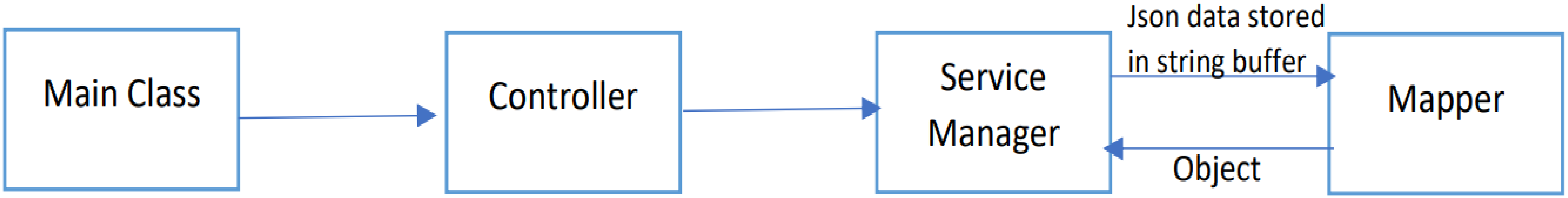
Service Manager

Data Object Mapper

MODEL

Database

Query Manager



Main Class

Controller

Service Manager

Mapper

Json data stored in string buffer

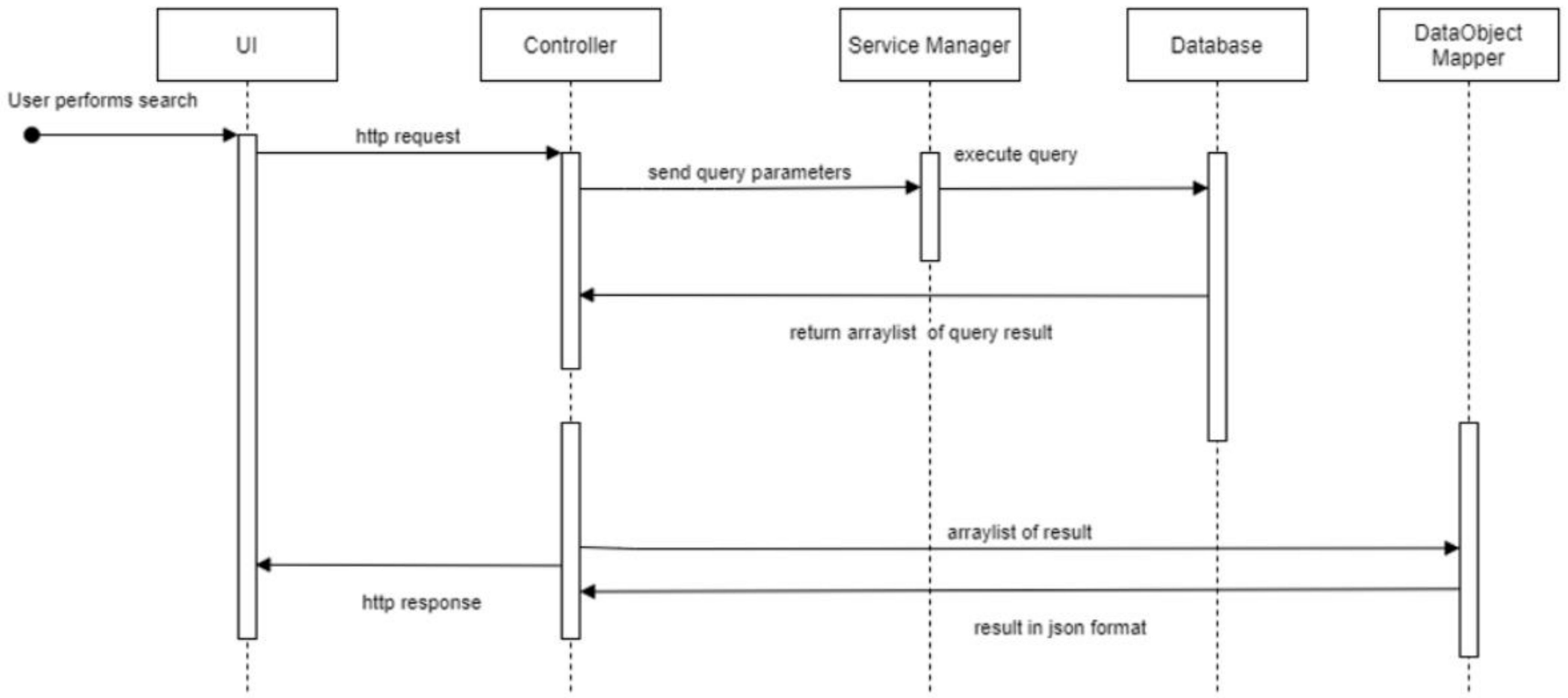
Object

Instantiates controller object

Gets the json data from the api via https

1) Establishes database connection.

2) Populates the database with the data after it is mapped.


UI
Controller
Service Manager
Database
DataObject Mapper
User performs search
http request
send query parameters
execute query
return arraylist of query result
arraylist of result
http response
result in json format

Development Process

Linear Process

- We decided to go with linear process as the requirements and objective of the project and deadline were clear.

Design patterns Used

Singleton pattern

We have used singleton pattern for creation of connection with our local Database.

Refactoring

Common methods are moved into interface which is then implemented by classes requiring the behaviour.

Data Source Architectural Pattern

Table Data Gateway

We have class called QueryManager in the Model Layer that handles search operations on the database.

Data Mapper

We have a DataMapper class that maps between json and java objects.

Service Manager

Acts as a mediator between Domain and data source

Object Relational Mapping

Vertical Mapping

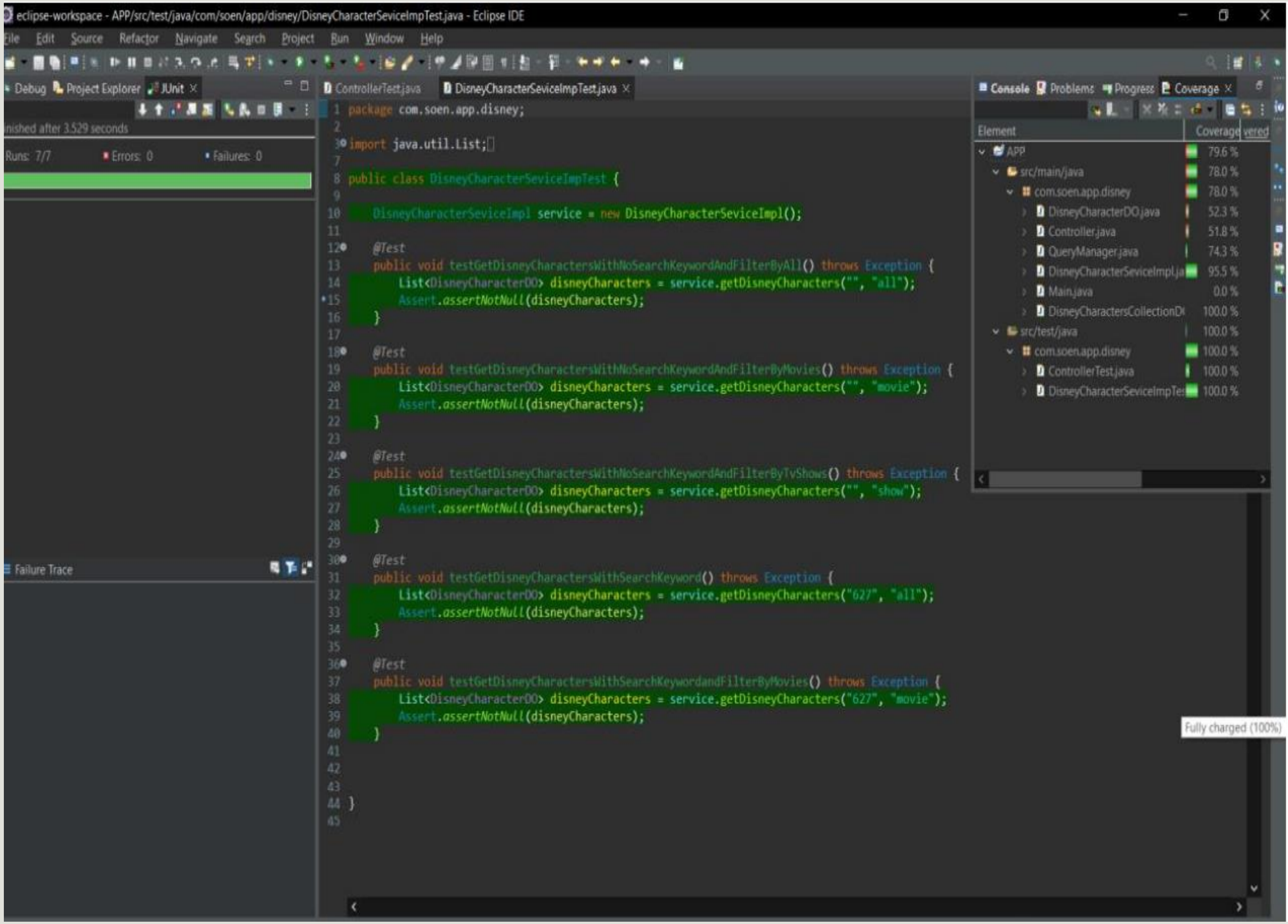
For model class in java we have table in our database.

Foreign key mapping

We have used foreign key mapping which serves as a link between our tables during query execution

Testing

JUNIT


eclipse-workspace - APP/src/test/java/com/soen/app/disney/DisneyCharacterSeviceImpTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Debug Project Explorer JUnit
ControllerTest.java DisneyCharacterSeviceImpTest.java
Console Problems Progress Coverage
Failure Trace
Fully charged (100%)

```
inished after 3.529 seconds
Runs: 7/7        Errors: 0        Failures: 0
```

```java
package com.soen.app.disney;

import java.util.List;

public class DisneyCharacterSeviceImpTest {

    DisneyCharacterSeviceImpl service = new DisneyCharacterSeviceImpl();

    @Test
    public void testGetDisneyCharactersWithNoSearchKeywordAndFilterByAll() throws Exception {
        List<DisneyCharacterDO> disneyCharacters = service.getDisneyCharacters("", "all");
        Assert.assertNotNull(disneyCharacters);
    }

    @Test
    public void testGetDisneyCharactersWithNoSearchKeywordAndFilterByMovies() throws Exception {
        List<DisneyCharacterDO> disneyCharacters = service.getDisneyCharacters("", "movie");
        Assert.assertNotNull(disneyCharacters);
    }

    @Test
    public void testGetDisneyCharactersWithNoSearchKeywordAndFilterByTvShows() throws Exception {
        List<DisneyCharacterDO> disneyCharacters = service.getDisneyCharacters("", "show");
        Assert.assertNotNull(disneyCharacters);
    }

    @Test
    public void testGetDisneyCharactersWithSearchKeyword() throws Exception {
        List<DisneyCharacterDO> disneyCharacters = service.getDisneyCharacters("627", "all");
        Assert.assertNotNull(disneyCharacters);
    }

    @Test
    public void testGetDisneyCharactersWithSearchKeywordandFilterByMovies() throws Exception {
        List<DisneyCharacterDO> disneyCharacters = service.getDisneyCharacters("627", "movie");
        Assert.assertNotNull(disneyCharacters);
    }



}
```

Element	Coverage	vered
APP	79.6 %	
src/main/java	78.0 %	
com.soen.app.disney	78.0 %	
DisneyCharacterDO.java	52.3 %	
Controller.java	51.8 %	
QueryManager.java	74.3 %	
DisneyCharacterSeviceImpl.ja	95.5 %	
Main.java	0.0 %	
DisneyCharactersCollectionD(	100.0 %	
src/test/java	100.0 %	
com.soen.app.disney	100.0 %	
ControllerTest.java	100.0 %	
DisneyCharacterSeviceImpTes	100.0 %	