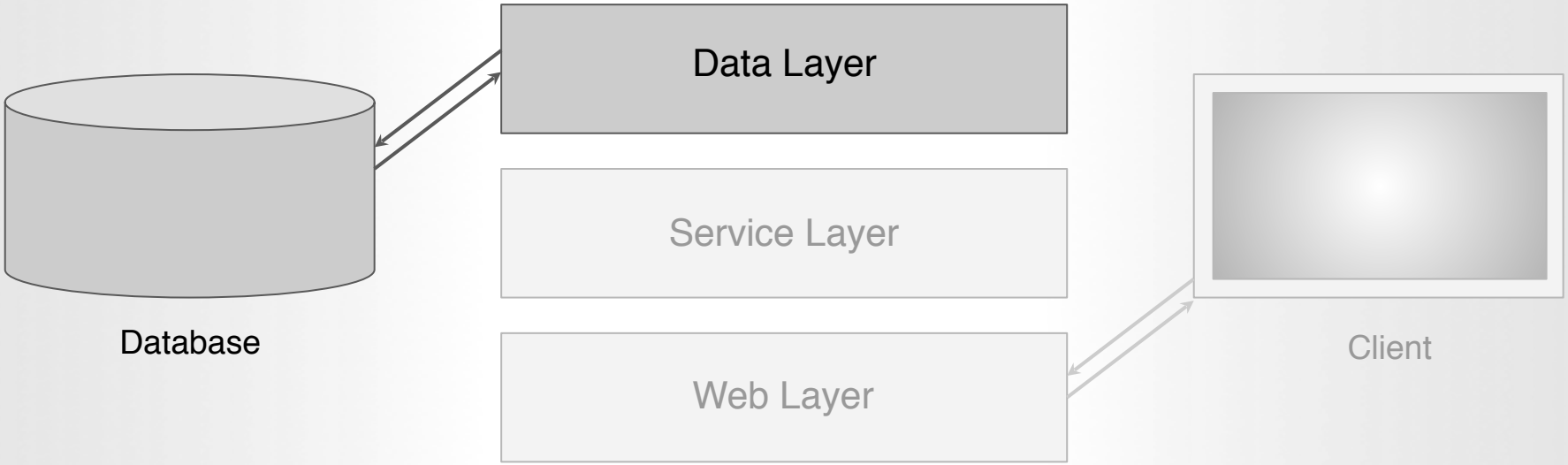
Stand Up

Project Preperations

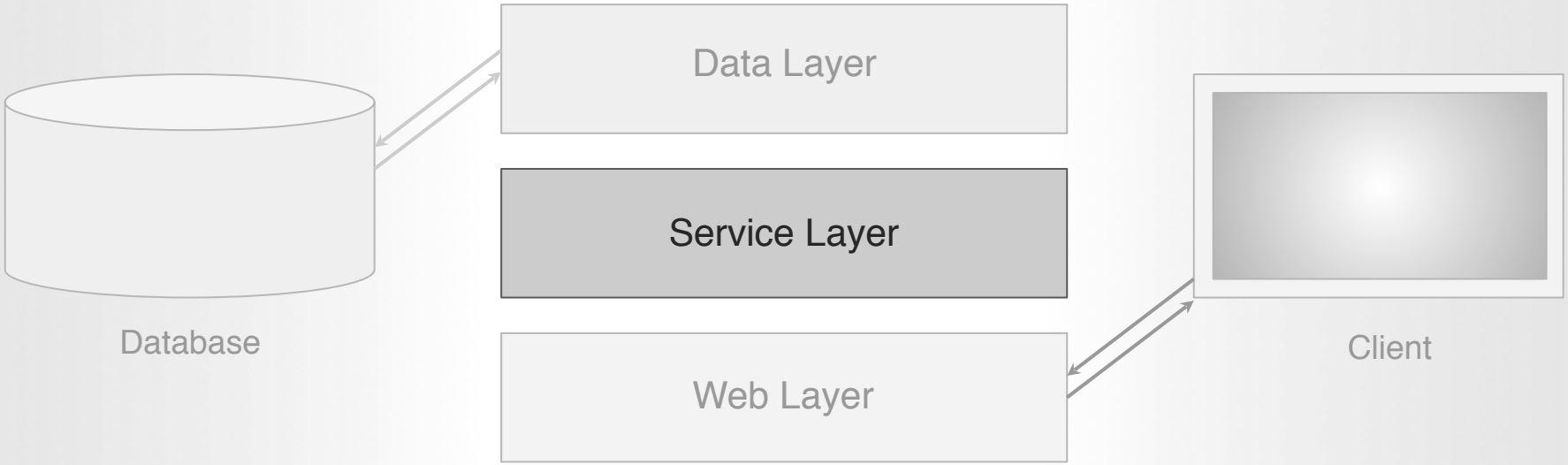
Project Preparations

We'll begin with the data layer.



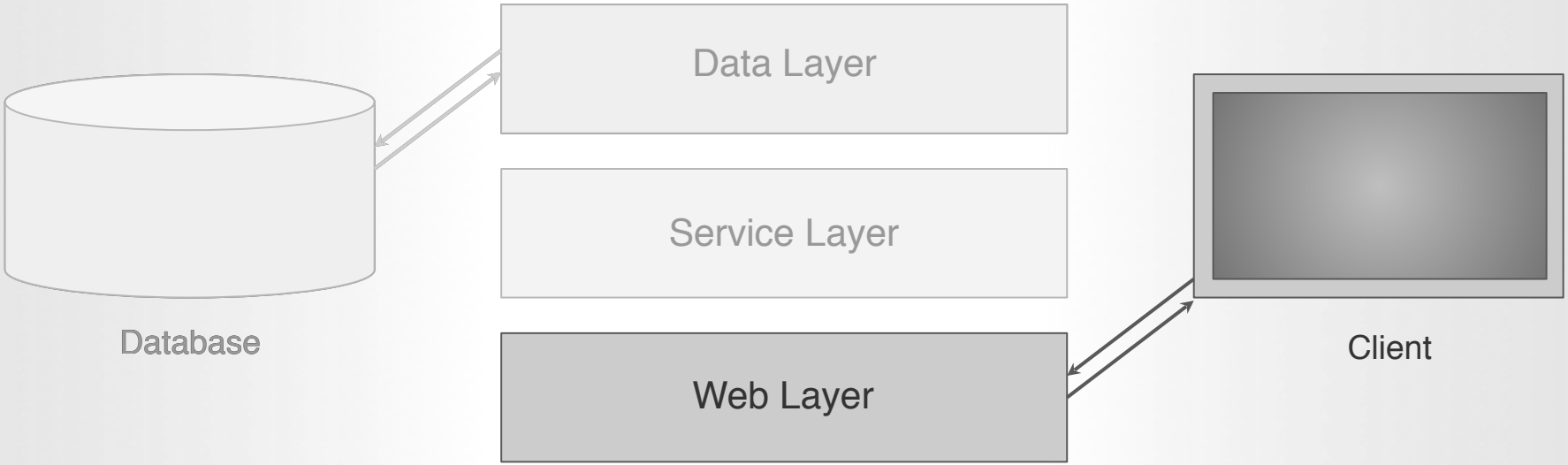
Project Preparations

Followed by the service layer.



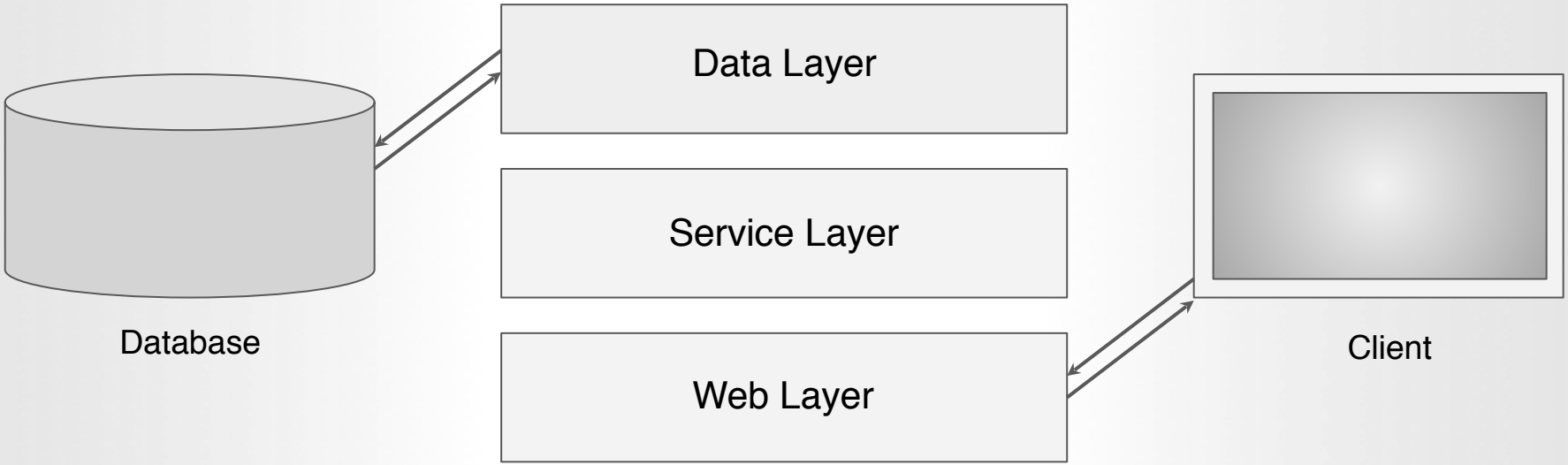
Project Preparations

And lastly, the web layer.



Project Preparations

We'll employee testing heavily throughout the process.



Project Preparations

We'll be making an application for a physical bookstore. This application will run on the computers at the help desk. The application should be built in a scalable and maintainable fashion.

Help Desk Personnel help customers looking for books. The customers generally are searching for books by title, author last name, author full name, or author last name and date range, so these are our top priorities.

The client mandates that we have 70-80% test coverage.

Project Preparations

Why do you think we might not want 100% test coverage?

Clients often are paying by the hour and testing has a point of diminishing returns. Testing every line of code means testing log output, DTOs, and other dumb components.

Generally, we should only test methods that have logic or may reasonably have logic in the future.

DTO

PAIR PRACTICE

It's time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Use Spring Initializer with the web, jpa, mysql, hateoas dependencies.
- Using the provided SQL statements, create and tables.
- Once the tables are created, seed them with the provided SQL statements.
- Create DTOs for each table.

Don't forget to update your application properties to connect to your database.

15 minutes!

Stay Seated &Take 3 Deep Breaths.

RELAX.

Now take a short walk. Clear your head. After a few minutes break, quickly review your notes. We'll start back in a few minutes.

Bidirectional Relationships

OBJECT RELATIONSHIPS

```
public class Director {
     @OneToMany(mappedBy="director", cascade=CascadeType.ALL, fetch=FetchType.EAGER)
     private Set<Movie> movies;
}


public class Movie {
     @ManyToOne(fetch=FetchType.EAGER)
     @JoinColumn(name="Directorid")
     private Director director;
}
```

Bidirectional Relationships

OBJECT RELATIONSHIPS

Bidirectional relationships allow JPA to be used to create more complex queries.

Using the relationships established in the previous slide, we could create a query to find movies by Director name and preferredGenre.

```
public interface MovieRepository extends JpaRepository<Movie, Integer> {

     List<Movie> findByDirectorNameAndDirectorGenre(String name, String genre);

}
```

Integration Tests

PAIR PRACTICE

It's time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Spend 10 minutes reviewing the DAO tests with your partner.
- What makes these integration tests and not unit tests?
- What exactly are the DAOs of this application expected to do?

10 minutes!

DAO

PAIR PRACTICE

It's time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Build your DAOs employing bidirectional relationships as needed to pass all data layer integration tests.

10 minutes!

Service Layer Mocks

PAIR PRACTICE

It's time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Build your Service Layer interface and mock it. DO NOT IMPLEMENT THE TEST CODE OR THE CODE TO PASS THE TESTS.

Remember the customers generally are searching for books by title, author last name, author full name, or author last name and date range (inclusively). We will need a method in the service layer for each of these.

15 minutes!

Stay Seated &Take 3 Deep Breaths.

RELAX.

Now take a short walk. Clear your head. After a few minutes break, quickly review your notes. We'll start back in a few minutes.

Service Layer Tests

PAIR PRACTICE

It's time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Write your service layer unit tests. DO NOT IMPLEMENT THE CODE TO PASS THE TESTS.

Remember the customers generally are searching for books by title, author last name, author full name, or author last name and date range (inclusively). We will need a method in the service layer for each of these.

15 minutes!

Service Layer

PAIR PRACTICE

It's time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Implement the code in the service layer to pass all tests.

Remember the customers generally are searching for books by title, author last name, author full name, or author last name and date range (inclusively). Which of these cannot be implemented by calling a single method from the DAO? These methods will need custom logic in the service layer. Recall that all business logic resides in the service layer.

15 minutes!

lunch.

Controller

PAIR PRACTICE

It's time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Implement the Web Layer (don't worry about tests for now)
- You'll need a route for each service layer method.
- When searching by author and year range, you should validate that the year is 4 digits and throw an exception if not.

20 minutes!

MockMvc

TESTING REST CONTROLLERS

MockMvc is a tool used to test REST controllers.

MockMvc creates the Spring framework tools available for testing purposes and mocks incoming requests.

MockMvc Tests

WATCH & LEARN

Close your laptop. Eyes on my screen. Pay attention.

Follow along as we walk through implementing a few tests with MockMvc

MockMvc Tests

PAIR PRACTICE

It's time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Finish writing unit tests for the remaining methods in the REST controller.
- Remember to test for Exception Handling. Lean on Google and past code to accomplish this.

25 minutes!