is a basic pattern where you separate the model (data), view (display), and controller (logic) into different files and directories.

1

represent knowledge. Could be a single object or a structure of objects.

2

A is a (visual) representation of its model

3

The is the link between the user and the system.

4

With the MVC Design Pattern, objects encapsulate the data, doesn't communicate directly with View & defines the logic, manipulation and processing of the data.

5

With the MVC Design Pattern, objects present the data and enables user interaction with it. It communicates with controller, notified about changes in data & controller notifies of any user-made changes.

6

What are the benefits of the MVC architecture?

7

What are the stages of UX design?

8

MVC architecture is a basic pattern where you separate the model (data), view (display), and controller (logic) into different files and directories.

1

Models represent knowledge. Could be a single object or a structure of objects.

2

A view is a (visual) representation of its model

3

The controller is the link between the user and the system.

4

With the MVC Design Pattern, Model objects encapsulate the data, doesn't communicate directly with View & defines the logic, manipulation and processing of the data.

5

With the MVC Design Pattern, View objects present the data and enables user interaction with it. It communicates with controller, notified about changes in data & controller notifies of any user-made changes.

6

Benefits fo the MVC architecture

- More reusable: use the same View for multiple application
- Easily adaptable: Each object has a clearly defined role. Good design principle

7

$Sketches \longrightarrow Wireframe \longrightarrow Mockups \longrightarrow Prototypes$

8

Why would you produce Mock-ups in the first place?

9

What are the 8 golden rules of user interface design?

10

In Spring model and [redacted] *used interchangeably.*

11

In the context of Spring, what is a ***Repository****?*

12

What are some important features of unit testing?

13

Integration Testing: Testing the system from end to end. What are the steps?

14

[redacted] *objects are passed around but never actually used. Usually they are just used to fill parameter lists.*

15

[redacted] *objects actually have working implementations, but usually take some shortcut which makes them not suitable for production.*

16

Strive for consistency
Seek universal usability
Offer informative feedback
Design dialogues that bring closure
Prevent errors
Permit easy reversal
Keep Users in control
Reduce short term memory load

8 golden rules

10

- Dialogue with customers; can be used to confirm requirements, show different choices & exchange ideas.
- Acts as a form of testing, by preventing misunderstanding and removing bugs early.

Reasons for Mockups

9

Data lives in a repository. Repositories are the Spring Mechanism for querying the underlying DB. We used CRUD (Create, Read, Update, Delete) repository.

12

In Spring model and entity used interchangeably.

11

Client submits a request to the web server. Web server maps request to a controller. Controller gets data via the DAO layer which gets data from DB. Controller passes data to view. View is processed. Web server sends view to client.

14

Tests dont build on other tests. They test one thing. Stay within class/process/network boundaries - dont test database as a side effect.

13

Fake objects actually have working implementations, but usually take some shortcut which makes them not suitable for production.

16

Dummy objects are passed around but never actually used. Usually they are just used to fill parameter lists.

15

provide canned answers to calls made during the test, usually not responding at all to anything outside what's programmed in for the test.

17

are pre-programmed with expectations which form a specification of the calls they are expected to receive. They can throw an exception if they receive a call they don't expect and are checked during verification to ensure they got all the calls they were expecting.

18

What is Spring Social?

19

An is a formal description of the behavior of a software product, expressed as a example or a usage scenario.

20

What are the benefits of ***Acceptance Tests****?*

21

is an architectural style that is the underlying architectural principal of the WWW. Clients can operate without knowing anything about & . Client and server must agree on the used.

22

What does ***RED****,* ***GREEN*** *&* ***REFACTOR*** *mean in the context of* ***Test Driven Development (TDD)****?*

23

Give some reasons why you would use ***Test Driven Development****?*

24

Mocks are pre-programmed with expectations which form a specification of the calls they are expected to receive. They can throw an exception if they receive a call they don't expect and are checked during verification to ensure they got all the calls they were expecting.

18

Stubs provide canned answers to calls made during the test, usually not responding at all to anything outside what's programmed in for the test.

17

An acceptance test is a formal description of the behavior of a software product, expressed as a example or a usage scenario.

20

Framework of Spring Boot; establishes connections between Spring boot apps and SaaS(Software as a service) providers e.g. Twitter, Facebook.
$SaaS = API + resources + Interface$

19

Representational State Transfer (REST) is an architectural style that is the underlying architectural principal of the WWW. Clients can operate without knowing anything about the server & the server's resources. Client and server must agree on the media type used.

22

Closer collaboration between developers and user/customer
Providing clear and unambiguous contract
Decrease chance and severity of defects

Acceptance tests benefits

21

- Quality Assurance becomes proactive rather than reactive.
- Estimations can be accurate enough to involve real customers in daily development.
- Short iterations. Each iteration produces a working product.
- Encourages good OO design practise.
- Encourages design for testability.
- Get an unambiguous progress meter.
- Build up a set of regression test as we go along.

Test Driven Development

24

- Write a test that does not work (RED)
- Make the test work (GREEN)
- Improve the code and eliminate duplication (REFACTOR)

Test Driven Development (TDD)

23