*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

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

*Benefits of the MVC architecture*

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  used interchangeably.*

11

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

12

*What is web app security mainly protecting?*

13

*In the context of web app, what does security mainly rely on?*

14

*A  is any potential occurrence, malicious or otherwise, that could harm an asset.*

15

*A  is a weakness that makes a threat possible.*

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

*Authentication and authorization. It also relies on auditing, confidentiality, integrity and availability.*

14

*Assets, which can either be tangible items such as a web page or a customer database, or intangible such as a company's reputation.*

13

*A vulnerability is a weakness that makes a threat possible.*

16

*A threat is any potential occurrence, malicious or otherwise, that could harm an asset.*

15

*An  is an action that exploits a vulnerability or enacts a threat.*

17

*Give examples of web app vulnerabilities.*

18

*What are the main causes of vulnerabilities?*

19

*What are the goals of Computer/network security hinges?*

20

*What is authentication?*

21

*Examples of authentication methods are?*

22

*What is authorization?*

23

- *: can access everything.*
- *: can access employee records and other relevant documents.*
- *: very limited access, usually don't need authentication.*

*The User Roles*

24

*Code injection: malicious executable code inserted into legitimate traffic sent to an endpoint.*
*Broken authentication and session management: compromising user identities in a variety of ways.*
*Cross-site scripting: similar to code injection, but involving scripts instead, drawn from inappropriate sources.*

18

*An attack is an action that exploits a vulnerability or enacts a threat.*

17

*Keeping unauthorized persons from gaining access to resources.*
*Ensuring that authorized persons can access the resources they need.*

20

*Poor programming in which exceptions, boundaries, credentials, etc., werent considered adequately.*

19

*An ID / password*
*A PIN (Personal Identification Number) code*
*An RFID card*
*Biometrics (usually considered the safest)*
*Fingerprints*
*Face Recognition*
*A USB token*
*A one-time password token*

*A lot of sites now use two-factor authentication*

22

*The process of recognizing a user's identity. It is the mechanism of associating an incoming request with a set of identifying credentials.*

21

*- Admin: can access everything.*
*- HR staff: can access employee records and other relevant documents.*
*- Guest: very limited access, usually don't need authentication.*

*The User Roles*

24

*Authorization is a security mechanism to determine access levels or user/client privileges related to system resources including files, services, computer programs, data and application features.*

23

*What are some important features of unit testing?*

25

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

26

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

27

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

28

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

29

*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.*

30

*What is Spring Social?*

31

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

32

*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.*

26

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

25

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

28

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

27

*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.*

30

*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.*

29

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

32

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

31

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

33

*Authentication =  +  .*

34

*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.*

35

*What are the two fundamental rules of TDD?*

36

*What are the mplications of TDD?*

37

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

38

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

39

*What are the limitations of TDD?*

40

*Authentication = connection + sign-in.*

34

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

*Acceptance tests benefits*

33

*Write new code only if a test has failed*
*Eliminate Duplication*

36

*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.*

35

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

***Test Driven Development (TDD)***

38

*Developers must write the unit tests*
*Development environment must provide quick feedback*
*System must consist of many loosely coupled components: Makes testing easy*
*Design must be organic: Running code provides feedback between decisions*

37

*Conceptually hard to start with*
*Can all programming tasks be driven by tests*
*- Security*
*- Concurrency*
*- Performance*

40

- *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*

39

*HTTP status codes: - 1xx:*
*- 2xx:*
*- 3xx:*
*- 4xx:*

41

*and*
*are guidelines focuesed on the communication aspects of the user interface*

42

*In the MVC architecture, what is a role of the model?*

43

*HTTP methods such as GET, PUT and DELETE are idempotent. What does this mean in practice?*

44

*What is the roll of an acceptance test?*

45

*Test-driven development can help ,*
*by providing a test suite that will pick up errors affecting existing functionality that are introduced by writing further production code.*
*Following a TDD process will not reduce , or ensure production code is error-free (unfortunately errors remain a hazard whatever method is used).*

46

*The specification by example process .*

47

*What is the earliest point at which you could start testing the layout of a user interface?*

48

*Design dialogues to yield closure and Provide informative feedback are guidelines focuesed on the communication aspects of the user interface*

42

*HTTP status codes: - 1xx: information*
*- 2xx: successful*
*- 3xx: redirection*
*- 4xx: client error*

41

*If an HTTP method is idempotent, it can be applied multiple times without changing the initial result.*

44

*Operations on the data occur in the model.*
*Business logic also happens in the model, but any display of this is dealt with in the view.*
*Mapping requests from the browser occurs in the controller.*

43

*Test-driven development can help reduce regression, by providing a test suite that will pick up errors affecting existing functionality that are introduced by writing further production code.*
*Following a TDD process will not reduce errors in test code, or ensure production code is error-free (unfortunately errors remain a hazard whatever method is used).*

46

*An acceptance test checks whether software meets functional requirements.*
*Whether the software is high in learnability would be assessed via a usability test.*
*A constraint refers to an aspect of the project that does not concern functionality, such as a release date, or the type of technology that must be used.*

45

*In a paper prototyping exercise.*
*Paper prototyping involves representing the user interface with pieces of paper that can be moved around, and is an early way of testing the format of the UI meets customer requirements. It is possible to test at all the other stages, but paper prototyping is the earliest.*

48

*The specification by example process reduces ambiguity in requirements capture.*

47

*What is a natural consequence of whole team responsibility?*

49

*If an API is designed according to RESTful architectural principles, the server must* ________________. *This helps the user (human or otherwise) navigate it without any prior knowledge of it.*

50

*The Create, Reuse, Update, Delete (CRUD) pattern may be used in* ________________. *Although system architecture involves design, this is at a much coarser level of granularity, and would not include the kinds of software design patterns that map directly to code.*

51

*If an API is designed according to RESTful architectural principles, the server must Provide responses that are self-descriptive.*
*This helps the user (human or otherwise) navigate it without any prior knowledge of it.*

50

*Developers have a broader knowledge of the code base. Whole team responsibility means that developers take ownership of the entire code base, rather than only working on specific parts of it.*

49

*The Create, Reuse, Update, Delete (CRUD) pattern may be used in The database and the user interface. Although system architecture involves design, this is at a much coarser level of granularity, and would not include the kinds of software design patterns that map directly to code.*

51