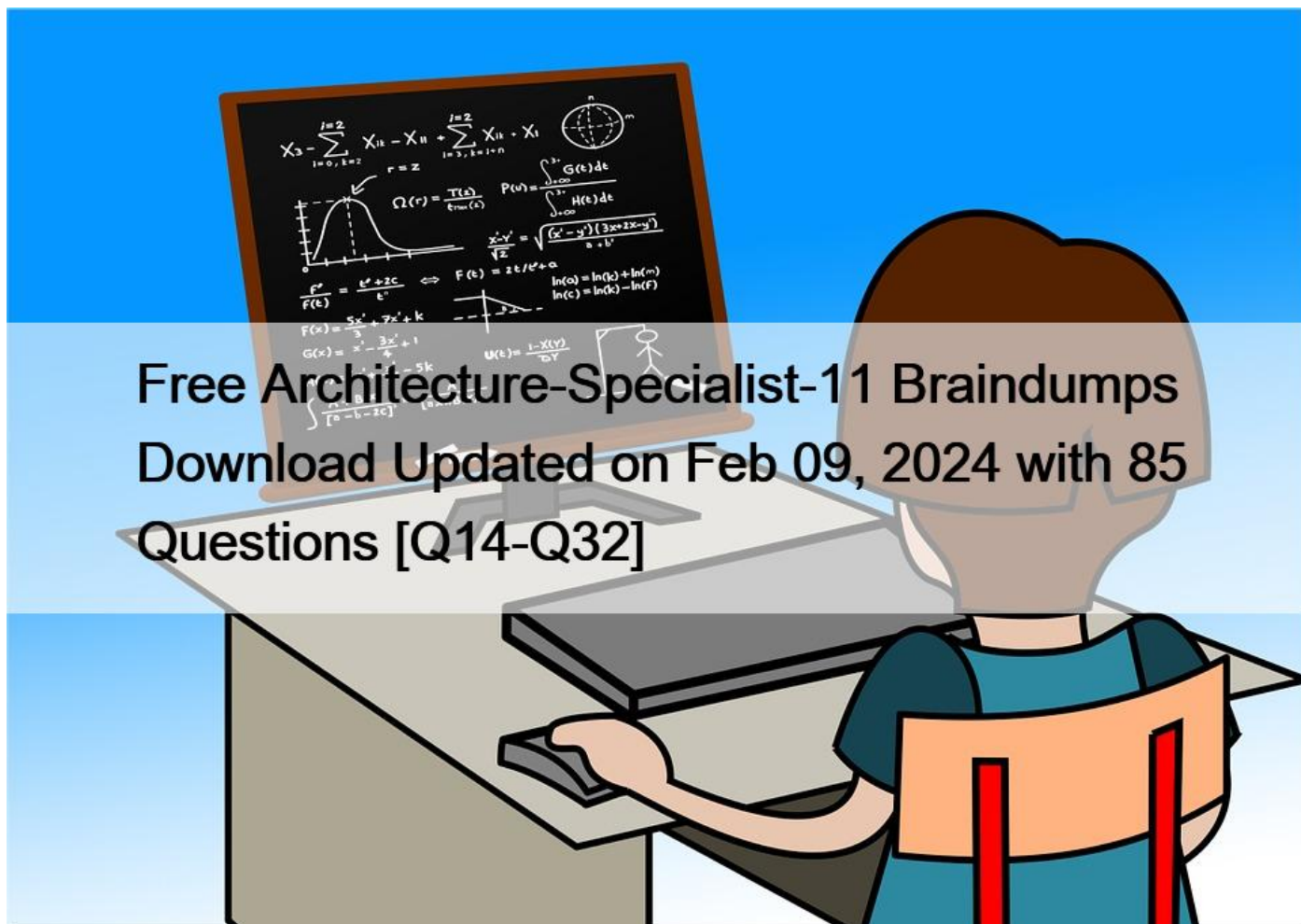


Free Architecture-Specialist-11 Braindumps Download Updated on Feb 09, 2024 with 85 Questions [Q14-Q32]



Free Architecture-Specialist-11 Braindumps Download Updated on Feb 09, 2024 with 85 Questions [Q14-Q32]

Free Architecture-Specialist-11 Braindumps Download Updated on Feb 09, 2024 with 85 Questions
OutSystems Architecture-Specialist-11 Exam Practice Test Questions

NEW QUESTION 14

What is a best practice for Mobile Application Architecture: Local Storage?

- * Do not use local storage(mobile device), instead try to use central storage(server side)
- * Map server side static entities into local entities
- * Have a local storage inside the CS Module instead of M_CS Module
- * Use a centralized local storage for all mobile applications to ensure maintainability

NEW QUESTION 15

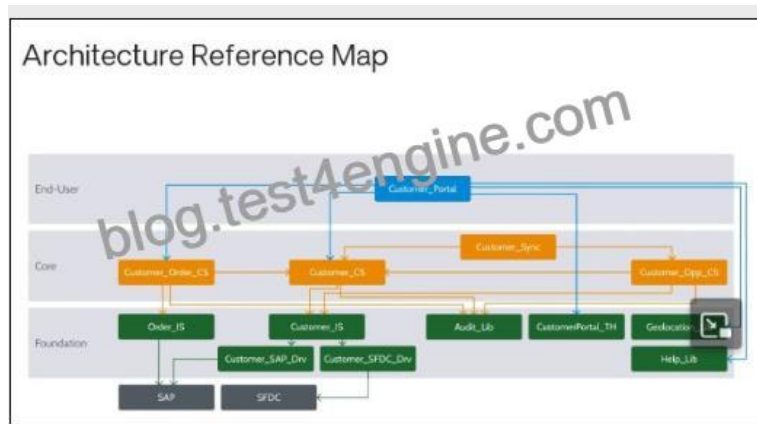
Which of the below is not part of the Architecture Validation Rules?

- * Layer Applications Correctly

- * Don't Mix Sponsors : Split the Apps based on the Lines of Business
- * Layer Entities Correctly
- * Layer Modules Correctly
- * Don't Mix Owners

NEW QUESTION 16

What is the Architecture Blueprint?



- * Is the end result of Architecture Design Process. At this point you should add dependencies between the Modules.
- * Above is the answer

NEW QUESTION 17

Which of the following is a benefit of having good architecture?

- * Poor service abstraction
- * Unmanageable dependencies
- * Manages complexity
- * Slow-moving legacy systems

NEW QUESTION 18

_Lib module is for

- * have several integration services with different systems, performing the same type of O operation (e.g. printers) you can create several drivers exposing the same API, with specialized implementations (like the transparency services pattern).
- * Technical wrapper to consume and normalize an external service
- * Reusable Core Services with public entities, actions, and blocks.
- * Theme, look & feel elements, menu, etc.
- * Generic Library module
- * Reusable UI Patterns for layout and display only – no Business logic.

NEW QUESTION 19

ISO/IEC 25010:2011 adds two more aspect to ISO/IEC 9126:1991. Which of the below is not part of it?

- * Compatibility : Compatibility was also added as a way to ensure that a piece of software can work together with other systems
- * Security : Security ensure software solutions can protect information and data

- * Integration : Integration to allow data exchange with other software

NEW QUESTION 20

Which of the below is not a best practice for mobile security:authentication?

- * Store password in local storage
- * Encrypt (only) sensitive data
- * Authentication : Use google or facebook for online authentication or fingerprint or pin for offline authentication

NEW QUESTION 21

Which of the below is NOT a disadvantage of having a fragmented system or microservices?

- * All of the above
- * Monitoring & Logging : for effective monitoring & logging, requires centralized service
- * Inter-process communication : network latency and hiccups
- * Complicated Debugging and troubleshooting : root cause may be deep inside the chain of services
- * Security : need to manage credentials and access management
- * Limited data mashup in memory and limited to APIs
- * Multiple transactions committed independently
- * Fault tolerance : communication errors, service consistency

NEW QUESTION 22

Which of the below is NOT a reason for adopting Naming Conventions for Modules

- * Reveal nature of each module
- * Ensure it belongs to the correct layer
- * Enforce the reference architecture
- * Normalize known patterns

NEW QUESTION 23

Of the options below, which one is a benefit of adopting the Architecture Canvas?

- * It's an automatic way to find and fix architecture issues.
- * It's a systematic approach to architecture design
- * It promotes the business users' collaboration and understanding
- * It's a faster architecture design

NEW QUESTION 24

Which of the below are FALSE about UI framework?

- * Patterns ; are blocks that provide common and reusable UI. Chat message, date picker and carousel are a few examples. Contains reusable UI components.
- * All of the above are true
- * Theme ; for the look and feel of your applications, which includes screen layouts, global stylesheet and grid definitions. Contains CSS styles and classes.
- * Templates ; Defines the structure of the application modules. Templates contains screen p definitions and basic set of functionalities, including authentication. Contains common layouts, menu, login and reference to the theme .

NEW QUESTION 25

_Th module is for

- * to have several integration services with different systems, performing the same type of operation (e.g. printers) you can create several drivers exposing the same API, with specialized implementations (like the transparency services pattern).
- * technical wrapper to consume and normalize an external service.
- * Theme, look & feel elements, menu, etc.
- * Generic Library module.
- * Reusable UI Patterns for layout and display only – no Business logic.

NEW QUESTION 26

Regarding Validation Rules for application composition, which of the following sentences is correct?

- * Common modules and services should be isolated in a separate app that can be referenced by other apps.
- * Owners of the Application should not be mixed, but mixing Sponsors has no direct impact on the release.
- * If you follow all the rules for validating modules, your application composition will be sound, and easy to maintain and deploy.
- * Applications in the Foundation layer can reference Applications in the End-user layer.

NEW QUESTION 27

Which of the below is NOT a reason why application Architecture is important when building quality software solutions?

- * Reduces Costs : Architecture benefits are not only technical, but affect how you operate
- * Reduces Risk : Architecture is a cost effective way to mitigate substantial risks and ensure success
- * Manages Complexity : Helps reduce emerging complexity by focusing on relevant properties and omitting irrelevant details, thus leading to simpler representation
- * Facilitate Change : When something comes up that requires you to veer from the original design or path, architecture of your system can show you exactly what needs to change and help with communication and planning.
- * Supports Planning : Helps the team anticipate and build solutions
- * Helps Communication : Helps you to get buy in from the business and to communicate overall plan to them
- * Drives Consensus : Helps create common ground with the team

NEW QUESTION 28

There are 3 common scenarios for Sharing a Style Guide. Which of the below is not part of the scenario.

- * Intranet (Single Sign On): Own Menu, Common Login Flow. Menu is defined in the Custom Template, but Login is defined in the Custom Theme. Application reference the Custom Them which picks up the Login.
- * Enterprise Apps: Common Menu, Common Login Flow. Menu is defined in the Custom Theme, but Login is defined in the Custom Template.
- * Independent Apps : Own Menu, Own Login Flow. Login and Menu is defined in the Custom Template. Applications reference to its own Application Theme thus do not use the Login and Menu in the Custom Template.
- * Portal : Common Menu, Common Login Flow. Login and Menu is defined in the Custom Them Application Theme reference the Custom Theme thus have a shared menu and login flow.

NEW QUESTION 29

Which of the below matches the most to Core Module Pattern – Base ECS Pattern…

- * … a wrapper used to contain the logic, actions and data that will expose code that is inside of external library or to inspect external database and import the data structures so they can be used as entities inside of OS
- * … caches only summary data that is frequently listed, joined or searched. Full detail for a single entry is fetched directly from external system. Use when whole database too big or costly to synchronize. Details are only required for single entities (not lists)

- * Same as ECS with local replica but synchronization logic is separated. Pro: Code independence. Consumers of CS is not affected by Sync. Sync can orchestrate several CS
- * … is a pattern with two modules, a connector module that can be used to encapsulate an external API with the input/output structures and a wrapper module to expose the normalized API to the consumers.
- * … is needed if data is coming from MULTIPLE external systems. IS will decide which driver to use depending on the data.
- * Same as ECS with local replica but API module is provided. So any changes to the external system can notify OS, which OS then gets update from the ERP system (subscription system)
- * Same as Base ECS pattern, but have a local replica. Store data to serve as a local cache. Pro: Leverage Entity Use, Simpler Integration API. Con: Less impact on source system
- * … Entity is not in OutSystems but in an external ERP system. IS just makes remote call to external system/database. No data is being kept inside OS. Data retrieval may not be optimized as it needs to traverse two different systems to get the information back. Con: Integration API must support all use cases
- * … tries to fetch data from local cache entity, if not there, get single entry from the external system. Cache only that record (read-through caching) Use when whole database too big or costly to synchronize. Integration only touches a small portion of the database. Avoid if access to lists of data is needed up front
- * … Entity is exposed as read-only and API is available to centralize business logic for entity creation/update

NEW QUESTION 30

In OutSystems, a Core Application can contain …

- * End-User, Core, and Foundation Modules.
- * Foundation and Core Modules
- * Only Foundation Modules.
- * End-User and Core Modules.

NEW QUESTION 31

Consider the common style guide scenarios presented in this course. Which of the following statements is true?

- * The “specialize a built-in Style Guide” scenario should be used when the changes to the base theme are not extensive.
- * The “clone a built-in Style Guide” should be used when you want to extend an existing theme.
- * Build-your-own Style Guide should be used when the changes in the theme are not extensive.

NEW QUESTION 32

Which of the below matches the most to Library Module Pattern – Extension Pattern…

- * … is a pattern with two modules, a connector module that can be used to encapsulate an external API with the input/output structures and a wrapper module to expose the normalized API to the consumers.
- * Same as ECS with local replica but API module is provided. So any changes to the external system can notify OS, which OS then gets update from the ERP system (subscription system)
- * … tries to fetch data from local cache entity, if not there, get single entry from the external system. Cache only that record (read-through caching) Use when whole database too big or costly to synchronize. Integration only touches a small portion of the database. Avoid if access to lists of data is needed up front
- * … is needed if data is coming from MULTIPLE external systems. IS will decide which driver to use depending on the data.
- * … caches only summary data that is frequently listed, joined or searched. Full detail for a single entry is fetched directly from external system. Use when whole database too big or costly to synchronize. Details are only required for single entities (not lists)
- * Entity is exposed as read-only and API is available to centralize business logic for entity creation/update
- * Same as ECS with local replica but synchronization logic is separated. Pro: Code independence. Consumers of CS is not affected by Sync. Sync can orchestrate several CS

- * Entity is not in Outsystems but in an external ERP system. IS just makes remote call to v external system/database. No data is being kept inside OS. Data retrieval may not be optimized as it needs to traverse two different systems to get the information back.
Con: Integration API must support all use cases
- * … a wrapper used to contain the logic, actions and data that will expose code that is inside of) external library or to inspect external database and import the data structures so they can be used as entities inside of OS
- * Same as Base ECS pattern, but have a local replica. Store data to serve as a local cache. Pro: Leverage Entity Use, Simpler Integration API. Con: Less impact on source system

Updated Verified Architecture-Specialist-11 dumps Q&As - Pass Guarantee or Full Refund:

https://www.test4engine.com/Architecture-Specialist-11_exam-latest-braindumps.html