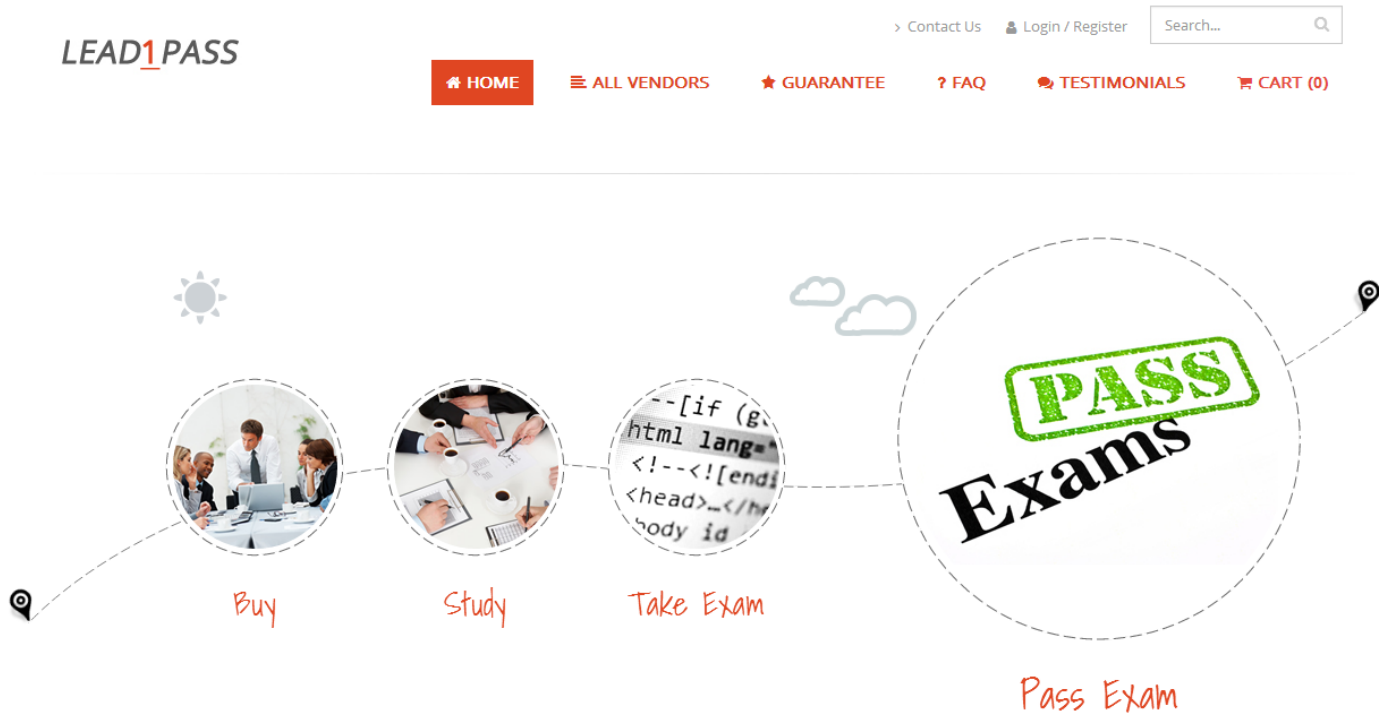
Lead1Pass



Try **PDF Demo** before you buy

Instant Download

After Payment, our system will send you the products you purchase in mailbox in a minute after payment. If not received within 2 hours, please contact us.

365 Days Free Updates

Free update is available within 365 days after your purchase. After 365 days, you will get 50% discounts for updating.

Money Back Guarantee

Full refund if you fail the corresponding exam in 60 days after purchasing. And Free get any another product.

Security & Privacy

We respect customer privacy. We use McAfee's security service to provide you with utmost security for your personal information & peace of mind.

http://www.lead1pass.com/

Latest Exam Guide & Learning Materials

Exam : **ACD300**

Title : Appian Certified Lead Developer

Vendor : Appian

Version : DEMO

NO.1 You have created a Web API in Appian. with the following URL to call it: https://exampleappiancloud.com/suite/webapi/usef_managefnent/ users ?username=)=john.smith. Which is the connect syntax forreferring to the user name parameter'

A. httpirequest usees username

B. httpirequest queryParameters.username

C. httpirequest formData username

D. httpirequest.queryParameters users username

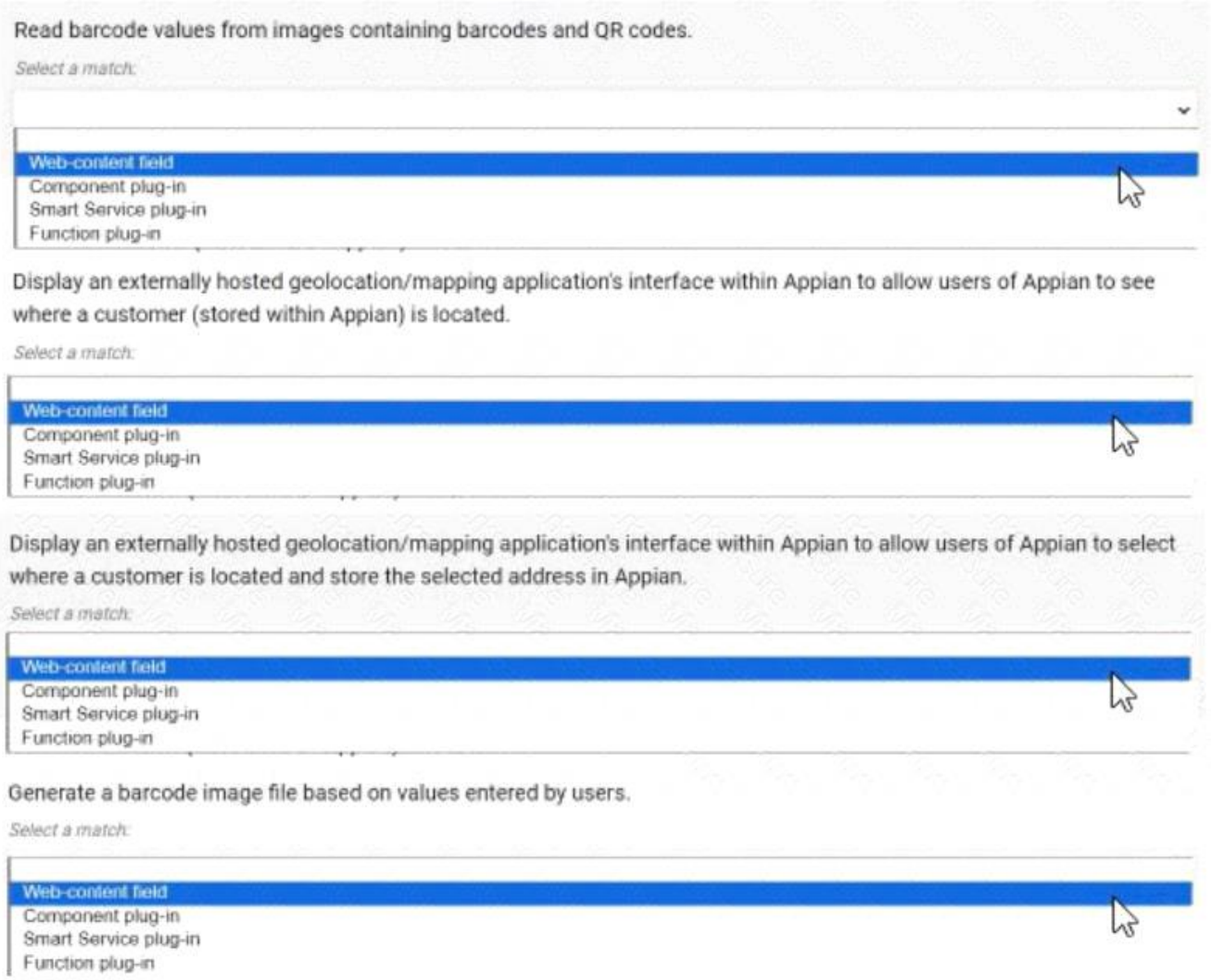
Answer: B

Explanation

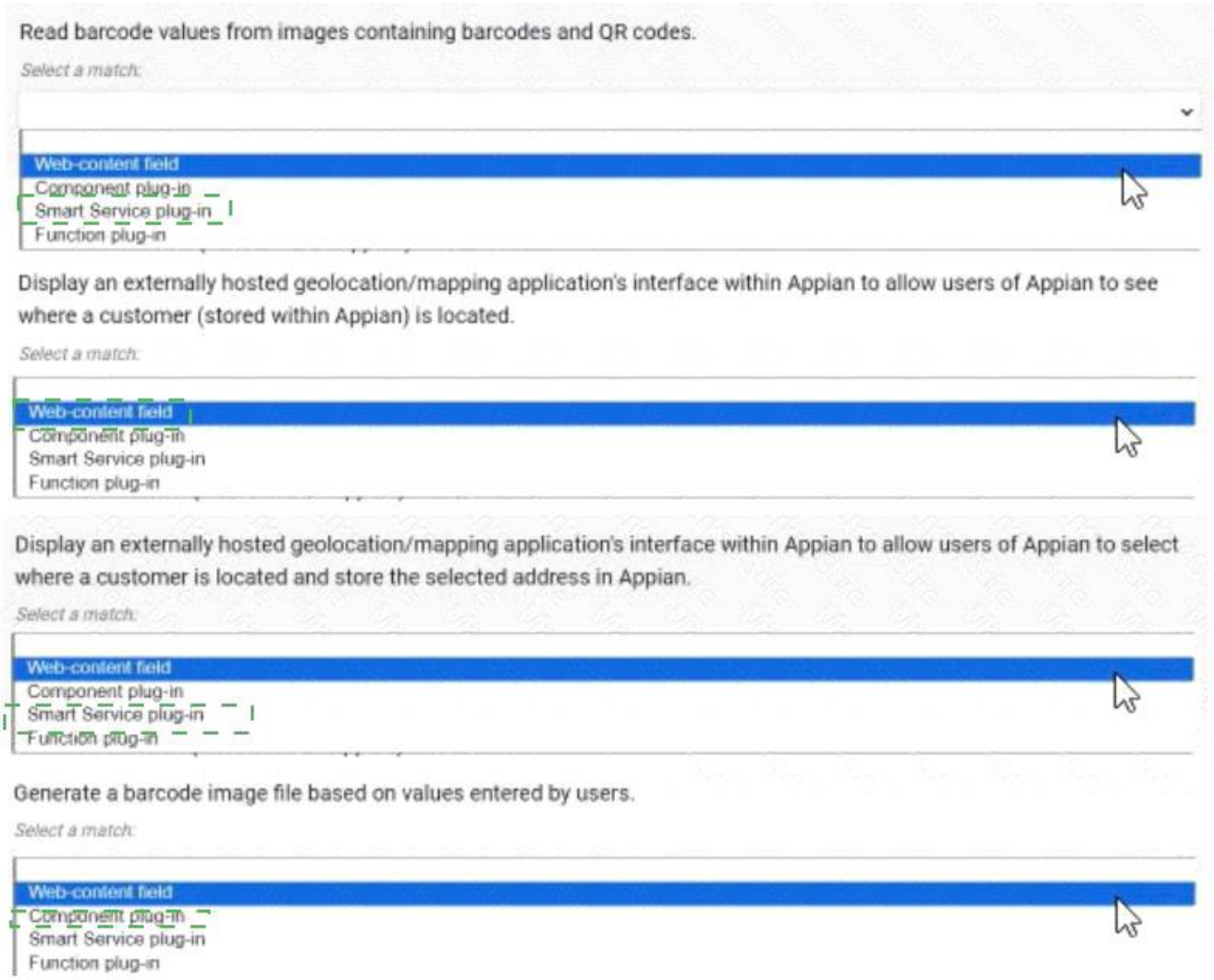
The correct syntax for referring to the username parameter in the Web API URL is httpirequest.queryParameters.username. This syntax allows you to access the value of the username parameter that is passed in the query string of the URL after the question mark (?). For example, if the URL ishttps://exampleappiancloud.com/suite/webapi/user_management/users?username=john.smith, then httpirequest.queryParameters.username will return john.smith. Verified References: Appian Documentation, section "Web API".

NO.2 For each requirement, match the most appropriate approach to creating or utilizing plug-ins Each approach will be used once.

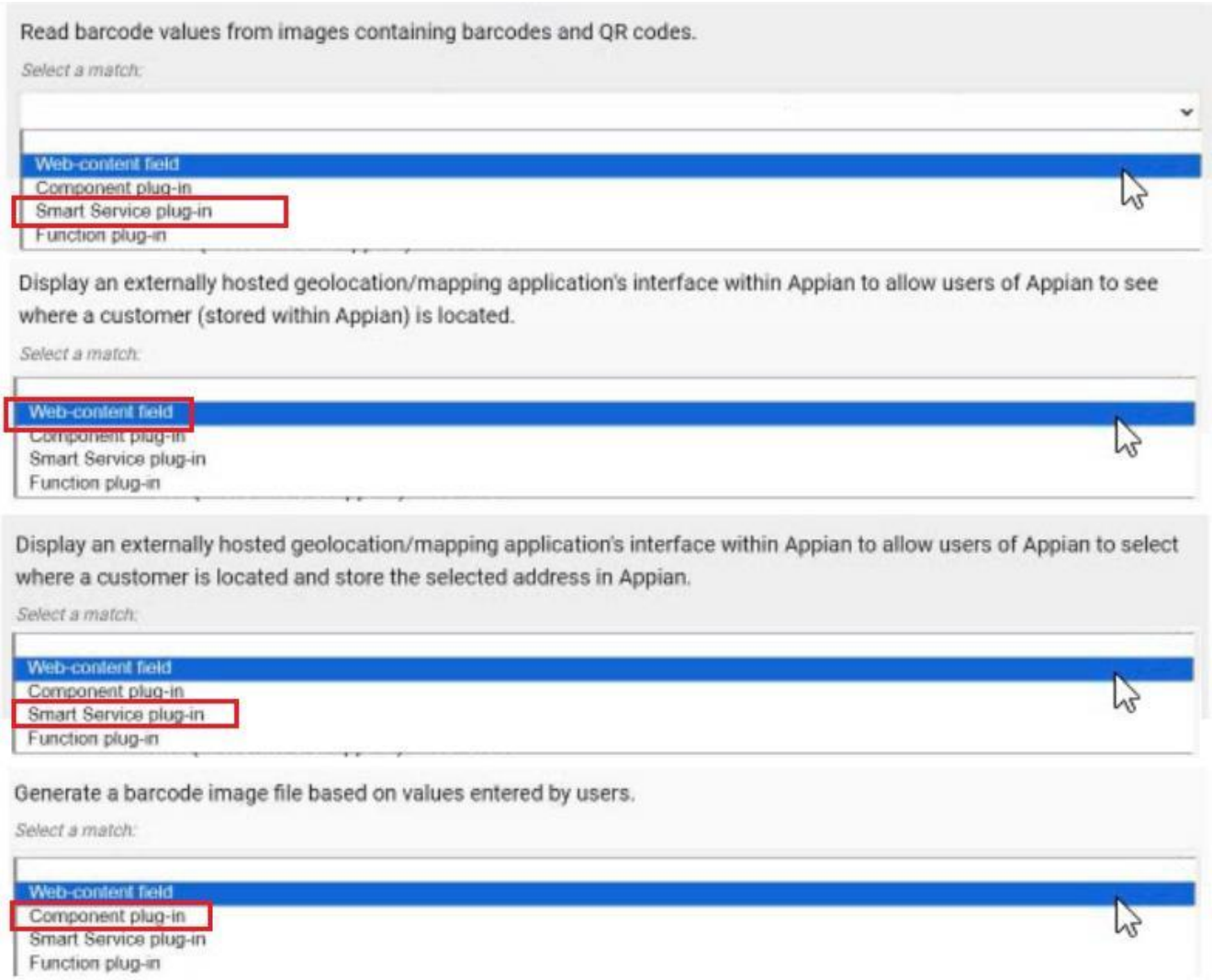
Note: To change your responses, you may deselect your response by clicking the blank space at the top of the selection list.


Read barcode values from images containing barcodes and QR codes.
Select a match:
Web-content field
Component plug-in
Smart Service plug-in
Function plug-in
Display an externally hosted geolocation/mapping application's interface within Appian to allow users of Appian to see where a customer (stored within Appian) is located.
Select a match:
Web-content field
Component plug-in
Smart Service plug-in
Function plug-in
Display an externally hosted geolocation/mapping application's interface within Appian to allow users of Appian to select where a customer is located and store the selected address in Appian.
Select a match:
Web-content field
Component plug-in
Smart Service plug-in
Function plug-in
Generate a barcode image file based on values entered by users.
Select a match:
Web-content field
Component plug-in
Smart Service plug-in
Function plug-in

Answer:



Explanation



Requirement: Read barcode values from images containing barcodes and QR codes. Correct approach: C.

Smart Service plug-in Exact explanation of correct approach taken from Appian Documentation: A smart service plug-in is a type of plug-in that allows you to create custom smart services that can be used in process models. A smart service can perform complex logic, interact with external systems, or manipulate data in Appian. A smart service plug-in can also leverage Java code to implement the functionality of the smart service. A smart service plug-in would be suitable for reading barcode values from images, as it can use Java libraries or APIs that can scan and decode barcodes and QR codes from image files. A smart service plug-in can also return the barcode values as outputs that can be used by other nodes or processes in Appian. A smart service plug-in can also be configured with input parameters, such as the image file, the barcode type, or the output format, that can customize the behavior of the smart service. A smart service plug-in can also have error handling and logging features that can handle any exceptions or failures that might occur during the barcode reading process.

Requirement: Display an externally hosted geolocation mapping applications interface within Appian to allow users of Appian to see where a customer (stored within Appian)is located. Correct approach: A. Web-content field Exact explanation of correct approach taken from Appian Documentation: A web-content field is a type of user interface component that allows you to display web content from an external source in a SAIL interface. A web-content field would be suitable for displaying an externally hosted geolocation mapping applications interface, as it can embed the web content in an

iframe and render it within the Appian interface.
You can also pass parameters to the web content, such as the customer's location, using the url parameter of the web-content field. A web-content field can also interact with other components in the Appian interface, such as buttons, grids, or forms, using the postMessage API. This way, you can create a seamless user experience that integrates the external geolocation mapping applications interface with the Appian functionality.
Requirement: Display an externally hosted geolocation mapping applications interface within Appian to allow users of Appian to select where a customer is located and store the selected address in Appian. Correct approach: A. Web-content field and C. Smart Service plug-in Exact explanation of correct approach taken from Appian Documentation: A web-content field and a smart service plug-in would be suitable for displaying an externally hosted geolocation mapping applications interface within Appian to allow users of Appian to select where a customer is located and store the selected address in Appian. A web-content field would be suitable for displaying the external geolocation mapping applications interface, as explained above. A smart service plug-in would be suitable for storing the selected address in Appian, as it can use Java code to receive the address data from the web content, validate it, and write it to a data store entity or a process variable.
Requirement: Generate a barcode image file based on values entered by users. Correct approach: B. Component plug-in Exact explanation of correct approach taken from Appian Documentation: A component plug-in is a type of plug-in that allows you to create custom user interface components that can be used in SAIL interfaces. A component plug-in can also leverage Java code to implement the functionality of the component. A component plug-in would be suitable for generating a barcode image file, as it can use Java libraries or APIs that can encode values into barcode formats and generate image files. A component plug-in can also display the barcode image file in the Appian interface and allow users to download or print it. A component plug-in can also interact with other components in the Appian interface, such as text fields, buttons, or forms, using the a!refreshVariable() function. This way, you can create a dynamic user experience that updates the barcode image file based on the values entered by users.

NO.3 You are planning a strategy around data volume testing for an Appian application that queries and writes to MySQL database.
You have administrator access to the Appian application and to the database.
What are two key considerations when designing a data volume testing strategy?

A. Data model changes must wait until towards the end of the protect.

B. Data from previous tests needs to remain in the testing environment prior to loading prepopulated data

C. Testing with the correct amount of data should be in the definition of done as part of each sprint.

D. large datasets must be loaded via Applan processes

E. The amount of data that needs to be populated should be determined by the project sponsor and the stakeholders based on their estimation

Answer: A,C
Explanation
When designing a data volume testing strategy for an Appian application that queries and writes to MySQL database, you should consider two key considerations:
* Testing with the correct amount of data should be in the definition of done as part of each sprint. Data volume testing is a type of testing that verifies how well an application performs when handling large amounts of data. Data volume testing is important to ensure that the application meets the

performance and quality requirements of the users and stakeholders. By including data volume testing in the definition of done as part of each sprint, you can ensure that each feature or functionality of your application is tested with realistic data volumes before being delivered to production. This way, you can identify and resolve any potential issues or bottlenecks early in the development cycle, and avoid any surprises or delays later on.
* Data model changes must wait until towards the end of the project. Data model changes are changes that affect the structure or schema of your database, such as adding, modifying, or deleting tables, columns, indexes, or constraints. Data model changes are risky and costly to make, especially when dealing with large amounts of data. Data model changes can affect the performance, functionality, or integrity of your
* application and database. Therefore, data model changes must wait until towards the end of the project, when you have finalized your requirements and design decisions, and have minimized your data volume testing efforts. By waiting until towards the end of the project to make data model changes, you can reduce the impact and complexity of those changes, and avoid any unnecessary rework or regression.
The other options are not as effective. Option A, data from previous tests needs to remain in the testing environment prior to loading prepopulated data, is not a key consideration for designing a data volume testing strategy, but rather a best practice for preparing your testing environment. Option B, large datasets must be loaded via Appian processes, is not a key consideration for designing a data volume testing strategy, but rather a technical implementation detail that may or may not be suitable for your application. Option C, the amount of data that needs to be populated should be determined by the project sponsor and the stakeholders based on their estimation, is not a key consideration for designing a data volume testing strategy, but rather an input or assumption that you need to validate before conducting your data volume testing.

NO.4 You are the lead developer for an Appian project, in a backlog refinement meeting You are presented with the following user story.
As a restaurant customer. I need to be able to place my food order online to avoid waiting in line for take out.' Which two functional acceptance criteria would you consider 'good'?
A. The system mutt handle up to 500 unique orders per day
B. The user cannot submit the form without filling out all required fields.
C. The user will receive an email notification when their order is completed.
D. The user will click Save, and the order information will be saved in the ORDER table and have audit history
Answer: B,C
Explanation
Functional acceptance criteria are the conditions that a user story must satisfy to be accepted by the user or stakeholder. They should be specific, measurable, achievable, relevant, and testable. In this case, two functional acceptance criteria that would be considered 'good' are:
* The user will receive an email notification when their order is completed. This is a specific, measurable, achievable, relevant, and testable criterion that describes a feature that the user needs to be informed of their order status.
* The user cannot submit the form without filling out all required fields. This is a specific, measurable, achievable, relevant, and testable criterion that describes afeature that the user needs to provide valid and complete information for their order.
The other options are not as good. Option A, the user will click Save, and the order information will

be saved in the ORDER table and have audit history, is not a functional acceptance criterion, but rather a technical implementation detail that is not relevant or visible to the user. Option C, the system must handle up to 500 unique orders per day, is not a functional acceptance criterion, but rather a non-functional requirement that describes a performance or quality attribute of the system.