Requirements Analysis

Smarter Balanced Assessment Consortium
Test Delivery System

Component: Test Delivery System
Purpose: Smarter Balanced Task Order 17, Administrative TDS DB Interface

Revision History

Revision Description	Author / Editor	Date
0.1 - First Draft	Rami Levy	Sept. 18, 2015
0.2 – Second draft – added design and risks. Added definition of procedure parameters including default values and population specification. Updated Figures 2 and 3 – main screen mockups.	Rami Levy	Nov. 18, 2015
1.0 – Final draft, incorporated feedback from Brandt that selection of one kind of SSID is required. Selection of the Session ID is optional. Updated Sections 1.2, 2, and 3.6.	Rami Levy	Nov. 23, 2015
1.1 – Update to clarify search field usage. Changes made to section 1.2, 2 (req. 6), and 3.6. Added missing requirement 7 in Table 1 and added Figure 4.	Rami Levy	Dec. 1, 2015
1.2 – Correct "Permissions" to "Permeability" in 1.1.	Rami Levy	Dec. 17, 2015
1.3 – Add warning about Student Report Processor requirements in section 3.7, Table 2. Updated Table 1, requirement 17, to add detail. Updated section 3.2 to clarify authorization.	Rami Levy	Apr. 19, 2016
1.4 – Remove `Firstname Lastname` from requirement to be placed in the log. Email address is unique and sufficient to identify the user.	Rami Levy	Apr. 20, 2016
1.5 – Additional access information added to section 3.2. Additional clarity added to configurable timeout parameter in Table 1, requirement 17.	Rami Levy	Apr. 21, 2016

Contents

1 Background 3
1.1 Problem Description 3
1.2 Proposed Solution 3
2 Requirements 4
3 Design 10
3.1 Single Sign-on Login 10
3.2 Authorization 10
3.3 Program Management Integration 10
3.4 Monitoring and Alerting Integration 10
3.5 User Interface 11
3.6 Search for Opportunity 11
3.7 Procedure Parameters and Defaults 13
4 Risks 17

Figures

Figure 1: Login Screen 10
Figure 2: Main Screen Mockup, Non-Segment-Related Procedures 11
Figure 3: Main Screen, Segment-Related Procedures 12
Figure 4: Possible Search Combinations 12

Tables

Table 1: Analysis of Task Order 17 Requirements 9
Table 2: Procedure Definitions and Parameters 16
Table 3: Risks 17

1 Background

There are certain administrative functions that occasionally need to be performed on the Test Delivery System (TDS) database, post-delivery. These functions include extending an opportunity's grace period, invalidating an opportunity, and reopening an opportunity. This task order seeks to create and provide a user interface for these stored procedures.

1.1 Problem Description

There are seven TDS administrative procedures that need to be provided, including: Set Opportunity Segment Permeability, Restore Test Opportunity, Reopen a Test Opportunity, Invalidate a Test Opportunity, Extend an Opportunity's Grace Period, Alter an Opportunity's Expiration, and Set Opportunity Segment Permissions. These need to be managed and executed via a user interface.

1.2 Proposed Solution

Smarter Balanced Task Order 17 commissions the creation of a user interface for the TDS administrative procedures. The solution will consist of the following changes:

- Create APIs in TDS to implement each of the seven stored procedures
- Create a new Java web application, with the same look & feel as standard Smarter apps, which provides a front-end for executing those administrative procedures
- App allows search for opportunity by External SSID or SSID, and/or Session ID
- App provides ability to execute any of the existing Stored Procedures (SPs) on an opportunity

2 Requirements

Task Order 17 requirements are as follows:

1. Application shall authenticate users using the same SSO system as the associated TDS.
2. Application shall only authorize users with a role that has appropriate permissions. Unauthorized users will receive an error message indicating this.
3. Application shall integrate with Program Management for tenancy, subscription, configuration, and component information.
4. Application shall integrate with MNA for monitoring and logging of alerts. A clear record of what was done, when, and by whom, should be sent to MNA by the application.
5. Application shall have a similar look & feel to the standard Smarter apps.
6. Application's user interface shall provide the ability to search for an opportunity by External SSID or SSID, and/or Session ID. Matches must be exact. Only one type of SSID value may be entered, not both. At least one of the three selections must be made in order to perform a search.
7. Application shall provide search results in a tabular fashion with the ability to paginate through the results, sort on any column, and select one or more opportunity from the set. Select-All/None functionality is not to be provided.
8. Application's user interface shall provide the ability to Set Opportunity Segment Permeability on an identified opportunity. This is additive to Grace Period Extension. In segmented tests where the segment is impermeable, Grace Period Extension will reopen the test but will not allow the student to go back into a prior impermeable segment. Adding Change Segment Permeability to Grace Period Extension allows the student to go back and change responses in otherwise inaccessible segments.
9. Application's user interface shall provide the ability to Restore Test Opportunity on an identified opportunity. Restore undoes a Reset action. This is used for situations where a test was inadvertently reset.
10. Application's user interface shall provide the ability to Reopen a Test Opportunity on an identified opportunity. Reopen allows a student to complete a previously expired test opportunity.
11. Application's user interface shall provide the ability to Invalidate a Test Opportunity on an identified opportunity. Invalidate makes an existing test opportunity invalid. This is used for situations such as when cheating is suspected.
12. Application's user interface shall provide the ability to Extend an Opportunity's Grace Period on an identified opportunity. If a test is paused beyond the grace period through no fault of the student, Grace Period Extension allows the test to be resumed as if the test was paused and resumed within the grace period.
13. Application's user interface shall provide the ability to Alter an Opportunity's Expiration on an identified opportunity. For test opportunities that have already expired, Extend Expiration allows the student to resume the test and be able to go back and change previous responses.
14. Application's user interface shall provide the ability to Reset a test Opportunity on an identified opportunity. Reset wipes out an existing student's test opportunity, allowing the student to take

an opportunity again. Reset is used for situations where the student inadvertently commenced an opportunity or inadvertently submitted a test too early.

15. Application shall visually provide the results of the execution of any of the administrative procedures when the process is completed. At a minimum, this shall consist of an indication of success or failure, along with an error message if applicable.
16. Application shall provide the ability for the user to log out of the current session for security reasons.
17. Application shall expire the session after 15 minutes of inactivity for security reasons.

The following table analyzes the requirements and provides a description of the technical approach towards addressing the requirement. This includes changes based on further discussions with Smarter Balanced.

Req. No.	Requirement	Task Description / Approach
1	Application shall authenticate users using the same SSO system as the associated TDS.	Integrate shared SSO mechanisms
2	Application shall only authorize users with a role that has appropriate permissions. Unauthorized users will receive an error message indicating this.	Integrate shared SSO mechanisms and Permissions
3	Application shall integrate with Program Management for tenancy, subscription, configuration, and component information.	Ensure user belongs to tenant whose subscription exists, is in good standing, and has not expired. Read configuration information from Program Management configuration.
4	Application shall integrate with MNA for monitoring and logging of alerts. A clear record of what was done, when, and by whom, should be sent to MNA by the application.	Integrate with MNA
5	Application shall have a similar look & feel as standard Smarter apps.	Use existing CSS
6	Application's user interface shall provide the ability to search for an opportunity by External SSID or SSID, and/or Session ID. Matches must be exact. Only one type of SSID value may be entered, not both. At least one of the three selections must be made in order to perform a search.	Text entry boxes, include validation per specifications (allow only valid characters). If no opportunities are selected by the search, do not allow any procedures to be selected.
7	Application shall provide search results in a tabular fashion with the ability to paginate through the results, sort on any column, and select one or more opportunity from the set. Select-All/None functionality is not to be provided.	Table shall consist of the following columns: Allow selection of multiple opportunities but one set of parameters applied to all. Table columns will be AltSSID/Name/SessionID/Status of Opportunity/Test Name/Date Started/DateExpired/DateCompleted/DatePaused. If segment-related procedure is selected, show one test segment per line with no dates necessary.

8	Application's user interface shall provide the ability to Set Opportunity Segment Permeability on an identified opportunity. This is additive to Grace Period Extension. In segmented tests where the segment is impermeable, Grace Period Extension will reopen the test but will not allow the student to go back into a prior impermeable segment. Adding Change Segment Permeability to Grace Period Extension allows the student to go back and change responses in otherwise inaccessible segments.	1. Create a stored procedure with this ability, along with an API on TDS which executes this stored procedure on an identified opportunity. 2. Required parameters must be entered by the user in another screen (or popup) 3. Error conditions (invalid parameters, etc.) should display an appropriate error message in the UI. 4. API must be secured to only Administrator roles. 5. Document the API.
9	Application's user interface shall provide the ability to Restore Test Opportunity on an identified opportunity. Restore undoes a Reset action. This is used for situations where a test was inadvertently reset.	1. Create a stored procedure with this ability, along with an API on TDS which executes this stored procedure on an identified opportunity. 2. Required parameters must be entered by the user in another screen (or popup) 3. Error conditions (invalid parameters, etc.) should display an appropriate error message in the UI. 4. API must be secured to only Administrator roles. 5. Document the API.
10	Application's user interface shall provide the ability to Reopen a Test Opportunity on an identified opportunity. Reopen allows a student to complete a previously expired test opportunity.	1. Create a stored procedure with this ability, along with an API on TDS which executes this stored procedure on an identified opportunity. 2. Required parameters must be entered by the user in another screen (or popup) 3. Error conditions (invalid parameters, etc.) should display an appropriate error message in the UI. 4. API must be secured to only Administrator roles. 5. Document the API.

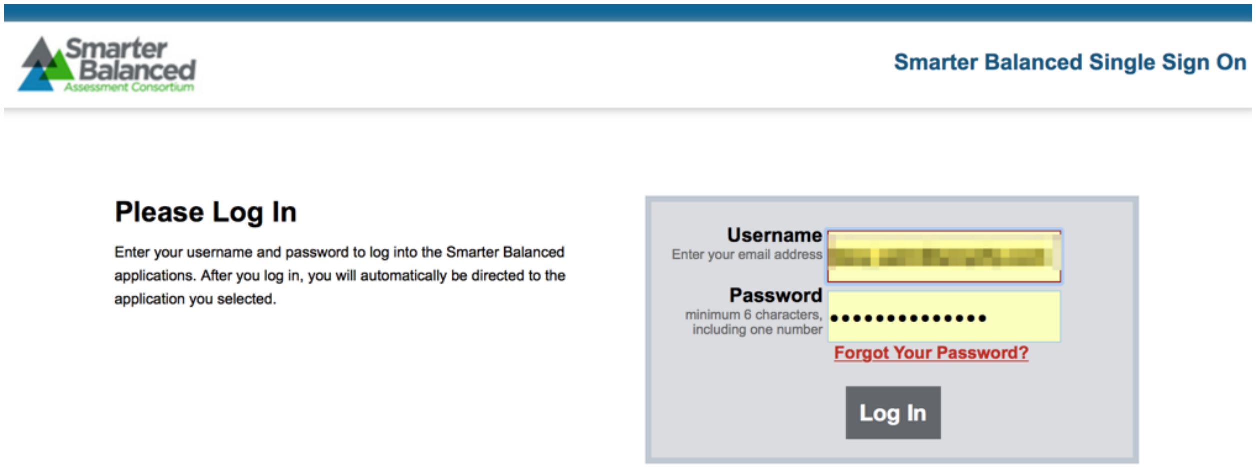
11	Application's user interface shall provide the ability to Invalidate a Test Opportunity on an identified opportunity. Invalidate makes an existing test opportunity invalid. This is used for situations such as when cheating is suspected.	1. Create a stored procedure with this ability, along with an API on TDS which executes this stored procedure on an identified opportunity. 2. Required parameters must be entered by the user in another screen (or popup) 3. Error conditions (invalid parameters, etc.) should display an appropriate error message in the UI. 4. API must be secured to only Administrator roles. 5. Document the API.
12	Application's user interface shall provide the ability to Extend an Opportunity's Grace Period on an identified opportunity. If a test is paused beyond the grace period through no fault of the student, Grace Period Extension allows the test to be resumed as if the test was paused and resumed within the grace period.	1. Create a stored procedure with this ability, along with an API on TDS which executes this stored procedure on an identified opportunity. 2. Required parameters must be entered by the user in another screen (or popup) 3. Error conditions (invalid parameters, etc.) should display an appropriate error message in the UI. 4. API must be secured to only Administrator roles. 5. Document the API.
13	Application's user interface shall provide the ability to Alter an Opportunity's Expiration on an identified opportunity. For test opportunities that have already expired, Extend Expiration allows the student to resume the test and be able to go back and change previous responses.	1. Create a stored procedure with this ability, along with an API on TDS which executes this stored procedure on an identified opportunity. 2. Required parameters must be entered by the user in another screen (or popup) 3. Error conditions (invalid parameters, etc.) should display an appropriate error message in the UI. 4. API must be secured to only Administrator roles. 5. Document the API.

14	Application's user interface shall provide the ability to Reset a test Opportunity on an identified opportunity. Reset wipes out an existing student's test opportunity, allowing the student to take an opportunity again. Reset is used for situations where the student inadvertently commenced an opportunity or inadvertently submitted a test too early.	1. Create a stored procedure with this ability, along with an API on TDS which executes this stored procedure on an identified opportunity. 2. Required parameters must be entered by the user in another screen (or popup) 3. Error conditions (invalid parameters, etc.) should display an appropriate error message in the UI. 4. API must be secured to only Administrator roles. 5. Document the API.
15	Application shall visually provide the results of the execution of any of the administrative procedures when the process is completed. At a minimum, this shall consist of an indication of success or failure, along with an error message if applicable.	Some modifications to the stored procedures are required in order to pass along the status information
16	Application shall provide the ability for the user to log out of the current session for security reasons.	Integrate shared SSO mechanisms
17	Application shall expire the session after 15 minutes of inactivity for security reasons.	Integrate shared SSO mechanisms. Application shall expire the session after a configurable number of minutes of inactivity for security reasons. • The value will be configurable via Program Management. • The default value is 15 minutes. • A value of zero will disable the login timeout and is not recommended. • Non-integer values are not permitted.

Table 1: Analysis of Task Order 17 Requirements

3 Design

3.1 Single Sign-on Login



Smarter Balanced Single Sign On

Please Log In

Enter your username and password to log into the Smarter Balanced applications. After you log in, you will automatically be directed to the application you selected.

Username
Enter your email address [redacted]

Password
minimum 6 characters, including one number ••••••••••••••

Figure 1: Login Screen

3.2 Authorization

Only authenticated users with roles having the following permissions shall be granted access to TDS Admin. These are configured via the Permissions application.

Role Name: TDS Administrator
Permissions Required: Opportunity Modify, Opportunity Read
Access Description: FULL – can search opportunities and execute all administrative procedures.

Role Name: TDS Admin viewer
Permissions Required: Opportunity Read
Access Description: READ ONLY – can only search opportunities.

IMPORTANT: The user with access to this system will be able to access any opportunity available in the TDS session database with which the system is associated, regardless of the user's hierarchy level. Since TDS is associated with one tenant at a time, cross-tenant interactions are not an issue.

3.3 Program Management Integration

The usual subscription management and configuration features shall be supported by TDS Administration.

3.4 Monitoring and Alerting Integration

The usual MNA integration features shall be supported by TDS Administration. Sufficient information shall be logged for audit purposes, including:

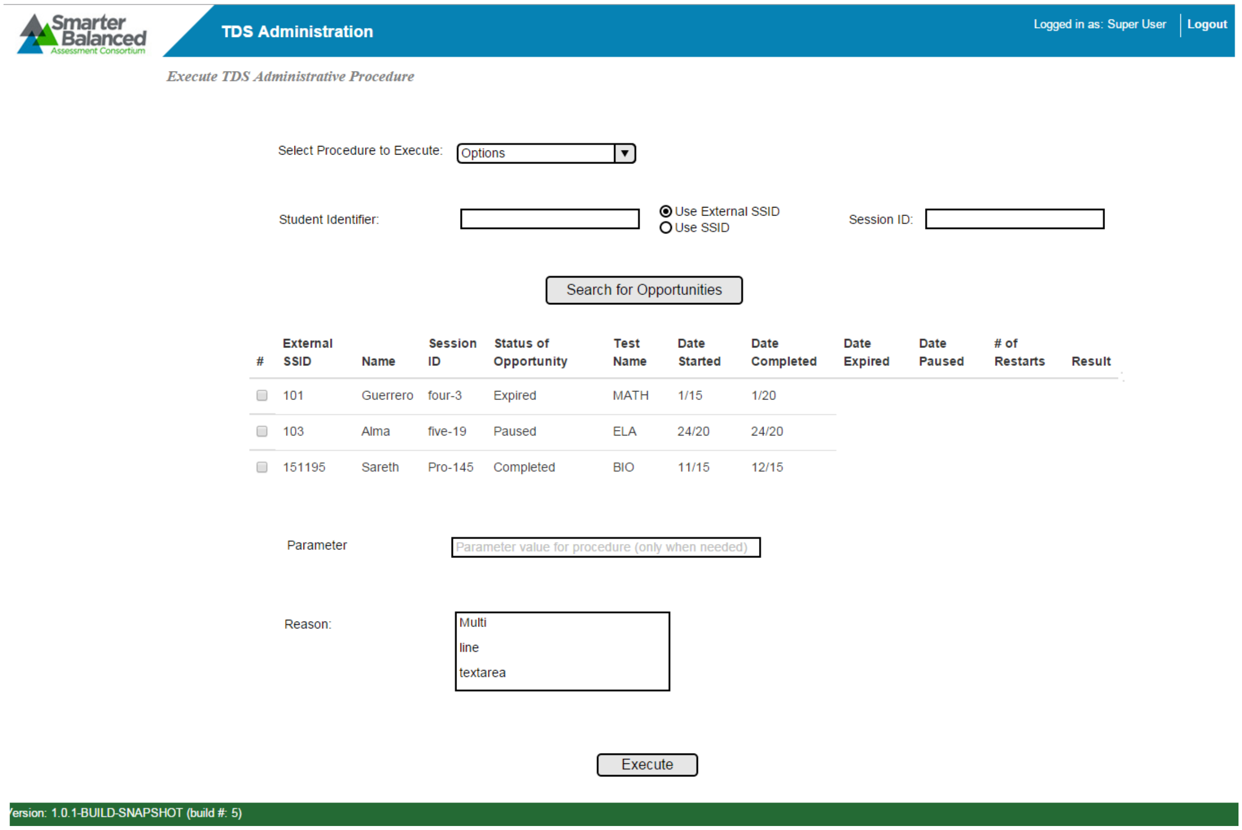
- User who has logged in
- Query that was executed

- Procedure that was executed including all relevant parameters, plus time/date and user's email, and the results of procedure execution for each opportunity (success/failure/reason).

3.5 User Interface

TDS Admin shall follow the same styling as existing Smarter Balanced applications.

3.6 Search for Opportunity



Execute TDS Administrative Procedure

Select Procedure to Execute: Options ▼

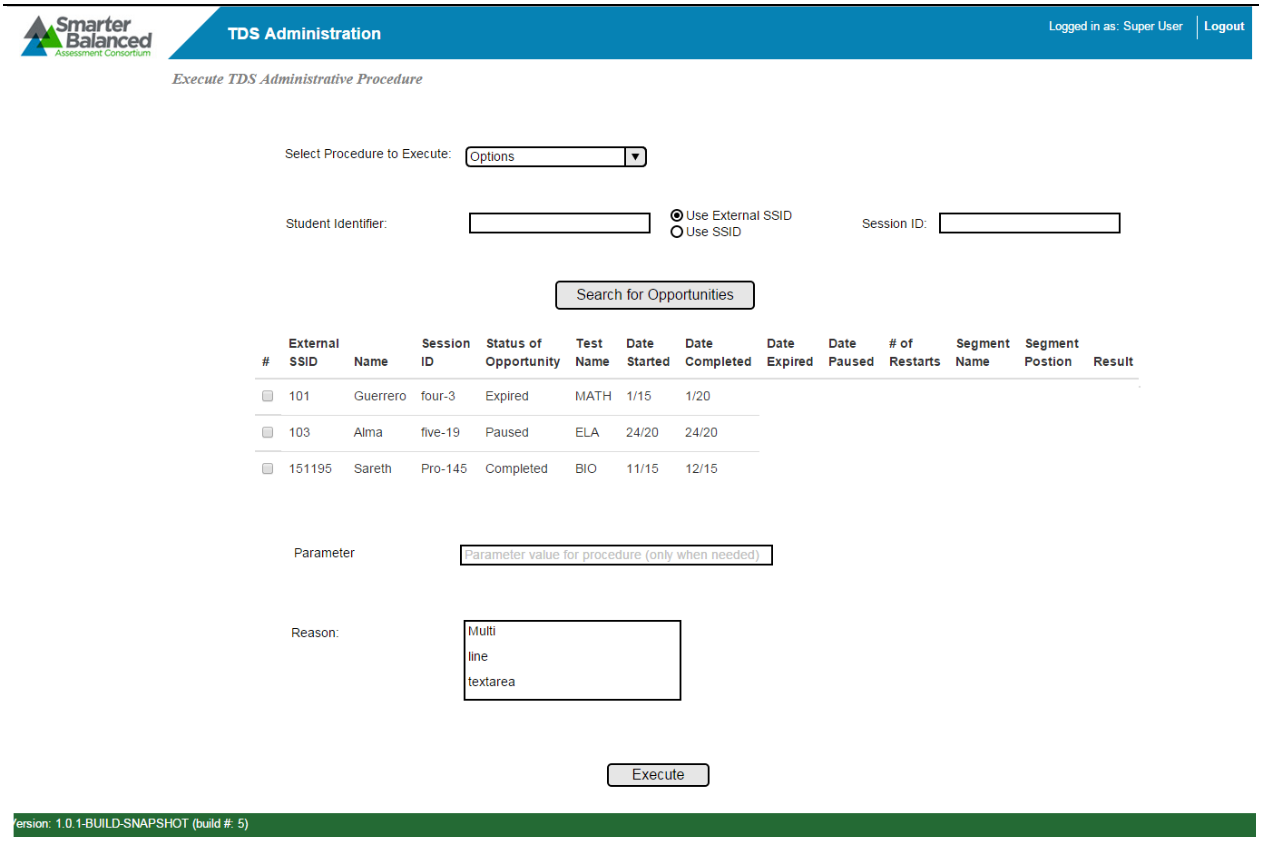
Student Identifier: ● Use External SSID ○ Use SSID Session ID:

#	External SSID	Name	Session ID	Status of Opportunity	Test Name	Date Started	Date Completed	Date Expired	Date Paused	# of Restarts	Result
□	101	Guerrero	four-3	Expired	MATH	1/15	1/20				
□	103	Alma	five-19	Paused	ELA	24/20	24/20				
□	151195	Sareth	Pro-145	Completed	BIO	11/15	12/15				

Parameter: Parameter value for procedure (only when needed)

Reason: Multi
line
textarea

Figure 2: Main Screen Mockup, Non-Segment-Related Procedures



Execute TDS Administrative Procedure

Select Procedure to Execute: Options ▼

Student Identifier: ● Use External SSID ○ Use SSID Session ID:

#	External SSID	Name	Session ID	Status of Opportunity	Test Name	Date Started	Date Completed	Date Expired	Date Paused	# of Restarts	Segment Name	Segment Postion	Result
□	101	Guerrero	four-3	Expired	MATH	1/15	1/20						
□	103	Alma	five-19	Paused	ELA	24/20	24/20						
□	151195	Sareth	Pro-145	Completed	BIO	11/15	12/15						

Parameter: Parameter value for procedure (only when needed)

Reason: Multi
line
textarea

Figure 3: Main Screen, Segment-Related Procedures

- Select the **procedure**(s) to execute:
  - Results and/or errors will be displayed on a per-opportunity basis.
  - Table column layout will depend on the procedure selected.
- **Search** fields:
  - External SSID or SSID. Only one may be used (optional)
  - Session ID (optional)
  - At least one of the three selections must be used.
  - Figure 4 shows the possible search combinations (1=value entered, 0=no value entered):

ALLOWED COMBINATION?	SSID	External SSID	Session ID
no	0	0	0
yes	0	0	1
yes	0	1	0
yes	0	1	1
yes	1	0	0
yes	1	0	1
no	1	1	0
no	1	1	1

Figure 4: Possible Search Combinations

- **Table** of results will support pagination and search:
  - Table column layout will depend on whether procedure requires segments to be shown.

  - If segments are not needed, columns will include External SSID/Name/Session ID/Status of Opportunity/Test Name/Date Started/Date Expired/Date Completed/Date Paused / # of Restarts. See Figure 2.
  - If segments are needed, one row per segment, grouped by test. Additional columns will include Segment Name, Segment Position, *v_ispermeable*. See Figure 3.
- Some procedures require additional user input **parameters** (see Table 2 for parameter details and defaults):
  - `Alter opportunity expiration` requires *v_dayincrement*.
  - `Set opportunity segment permeability` requires: *v_ispermeable*, and *v_restoreon*
  - `Extend Opportunity Grace Period` requires: *v_selectedsitting*.
- **Reason for Change:** An optional text entry field to describe the reason for making all changes.
- **Execute** button: Perform the requested functions.

3.7 Procedure Parameters and Defaults

Function	Description	Input Parameters and Default Values
Reset a_resettestopportunity	Reset wipes out an existing student's test opportunity, allowing the student to take an opportunity again. Reset is used for situations where the student inadvertently commenced an opportunity or inadvertently submitted a test too early. **Warning**: For proper processing of downstream events, the *Student Report Processor* must be in place, and its queue of opportunities should be empty before executing this procedure. Failure to do so may also affect the results of other procedures.	○ *v_oppkey*: test opportunity key value. A valid test opportunity key is a must. Value will be obtained from the user-selected items in the table. ○ *v_requestor*: this variable is for auditing purposes on who changed/modified a given opportunity through this procedure (100 char max). Value will be in the format `<email_address>`. Characters past the 100th will be truncated. ○ *v_reason*: this variable is also intended for auditing purposes to log the reason behind administratively changing a given opportunity. This will be populated by the optional text entry field "Reason".

Invalidate a_invalidatetestopportunity	Invalidate makes an existing test opportunity invalid. This is used for situations such as when cheating is suspected.	○ *v_oppkey*: test opportunity key value. A valid test opportunity key is a must. Value will be obtained from the user-selected items in the table. ○ *v_requestor*: this variable is for auditing purposes on who changed/modified a given opportunity through this procedure (100 char max). Value will be in the format `<email_address>`. Characters past the 100th will be truncated. ○ *v_reason*: this variable is also intended for auditing purposes to log the reason behind administratively changing a given opportunity. This will be populated by the optional text entry field "Reason".
Restore a_restoretestopportunity	Restore undoes a Reset action. This is used for situations where a test was inadvertently reset. **Warning**: For proper processing of downstream events, the *Student Report Processor* must be in place, and its queue of opportunities should be empty before executing this procedure. Failure to do so may also affect the results of other procedures.	○ *v_oppkey*: test opportunity key value. A valid test opportunity key is a must. Value will be obtained from the user-selected items in the table. ○ *v_requestor*: this variable is for auditing purposes on who changed/modified a given opportunity through this procedure (100 char max). Value will be in the format `<email_address>`. Characters past the 100th will be truncated. ○ *v_reason*: this variable is also intended for auditing purposes to log the reason behind administratively changing a given opportunity. This will be populated by the optional text entry field "Reason".
Reopen a_reopenopportunity	Reopen allows a student to complete a previously expired test opportunity.	○ *v_oppkey*: test opportunity key value. A valid test opportunity key is a must. Value will be obtained from the user-selected items in the table. ○ *v_requestor*: this variable is for auditing purposes on who changed/modified a given opportunity through this procedure (100 char max). Value will be in the format `<email_address>`. Characters past the 100th will be truncated. ○ *v_reason*: this variable is also intended for auditing purposes to log the reason behind administratively changing a given opportunity. This will be populated by the optional text entry field "Reason".

Grace Period Extension a_extendingoppgraceperiod	If a test is paused beyond the grace period through no fault of the student, Grace Period Extension allows the test to be resumed as if the test was paused and resumed within the grace period.	○ *v_oppkey*: test opportunity key value. A valid test opportunity key is a must. Value will be obtained from the user-selected items in the table. ○ *v_requestor*: this variable is for auditing purposes on who changed/modified a given opportunity through this procedure (100 char max). Value will be in the format `<email_address>.` Characters past the 100th will be truncated. ○ *v_selectedsitting*: valid values are 0 or greater. default value should be set to zero. Value populated by user entry. ○ *v_doupdate*: valid values are 1 or 0. Should be set to 1 to update the opportunity. When set to 0, it will check all the conditions and in the code and display the result without updating the opportunity. Set this to 1. ○ *v_reason*: this variable is also intended for auditing purposes to log the reason behind administratively changing a given opportunity. This will be populated by the optional text entry field “Reason”.
Extend Expiration a_alteropportunityexpiration	For test opportunities that have already expired, Extend Expiration allows the student to resume the test and be able to go back and change previous responses.	○ *v_oppkey*: test opportunity key value. A valid test opportunity key is a must. Value will be obtained from the user-selected items in the table. ○ *v_requestor*: this variable is for auditing purposes on who changed/modified a given opportunity through this procedure (100 char max). Value will be in the format `<email_address>`. Characters past the 100th will be truncated. ○ *v_dayincrement*: value should be equal to number of days a given opportunity needs to be extended by. Value is populated by user entry. ○ *v_reason*: this variable is also intended for auditing purposes to log the reason behind administratively changing a given opportunity. This will be populated by the optional text entry field “Reason”.

Change Segment Permeability a_setopportunitysegmentperm	This is additive to Grace Period Extension. In segmented tests where the segment is impermeable, Grace Period Extension will reopen the test but will not allow the student to go back into a prior impermeable segment. Adding Change Segment Permeability to Grace Period Extension allows the student to go back and change responses in otherwise inaccessible segments.	• *v_oppkey*: test opportunity key value. A valid test opportunity key is a must. Value will be obtained from the user-selected items in the table. • *v_requestor*: this variable is for auditing purposes on who changed/modified a given opportunity through this procedure (100 char max). Value will be in the format `<email_address>`. Characters past the 100th will be truncated. • *v_ispermeable*: flag value that indicates whether the segment is permeable or not. Value will be set based on user input. • *v_restoreon*: the variable represents what state the segment needs to be set to. Valid values include segment, paused, completed. Default value should be set to segment. Value will be set based on user input. ▪ 'segment': restore the default when the testee exits the segment (so he/she is allowed one visit into the segment. behaves most like the natural manner) ▪ 'paused': restore the default when the testee pauses the test. if testee does not visit the segment, it doesn't matter. ▪ 'completed': the default is never restored • *v_reason*: this variable is also intended for auditing purposes to log the reason behind administratively changing a given opportunity. This will be populated by the optional text entry field "Reason".

Table 2: Procedure Definitions and Parameters

4 Risks

Risk	Mitigation
It's possible that a large number of opportunities could be affected by a change, and the change is not reversible	• Only administrators should have access to this tool • Do not allow "select all" in list of opportunities.

Table 3: Risks