Component Test Plan

SBAC-11Test Registration

Version 1.6

Revision History

Date	Version	Description	Author
04/23/14	1.0	Initial Draft and introduction	Ryan Marinello
04/23/14	1.1	Requirement tracing	Ryan Marinello
04/23/14	1.2	Level 2's	Ryan Marinello
04/23/14	1.3	API	Ryan Marinello
04/23/14	1.4	Dependent Services	Ryan Marinello
04/23/14	1.5	Dependent Components	Ryan Marinello
04/23/14	1.6	Validations	Ryan Marinello
05/05/14	1.6	Table Updated	Ryan Marinello
05/05/14	1.6	Edits	Ryan Marinello

Smarter Balanced Assessment Consortium RFP #11	Version: 1.6
Component Test Plan	Date: 05/05/2014
Test Registration	

Contents

I. Purpose

The purpose of this document is to describe the testing strategy and test scenarios applied to the tested component and required dependent components.

I.   Overview

1.   The overall responsibilities of this component are:
    a.   Consuming REGISTRATION package Assessments
    b.   Interaction with Test Spec Bank
    c.   Provide eligibility to Data Warehouse
    d.   User Interface for Management of Information
    e.   Uploads, Previews, Validations, File Data
    f.   RESTful interface with Persistence ENUMs
    g.   Tenancy chain crosswalk
    h.   SSO Authorization
    i.   SSO Export
    j.   Test Delivery endpoints for Proctor and Student
    k.   Student Data Export
    l.   Provides Identity Information

The user audience for Test Registration component:
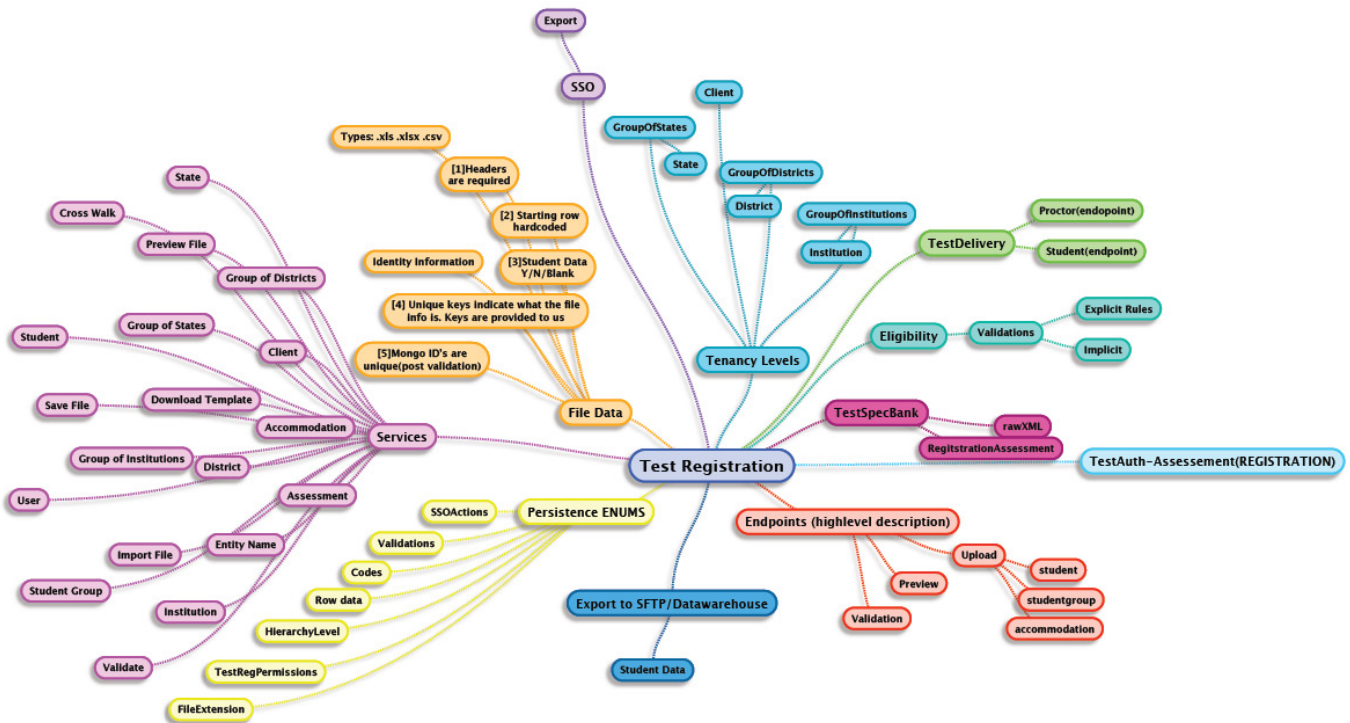1)   Users managing student data and assessment event data information.

II. Component: Test Registration

The Test Registration component registers student(s) for assessments and interacts with the Student Information System (SIS) aggregating student information, accessibility profile, and identity information that can be sent to a Data Warehouse by means of Test Delivery.

III. Scope of Testing

This plan focuses on verification and validation testing of the Test Registration component based on the Level 2 component requirements. Mission statement: prove level 2 requirements for Test Authoring which are believed to be correct. Heuristic: CBC (contextual, beneficial, congruent)

Scenario	Endpoints and examples
Display and Describe the Test Registration API RESTful Services and Controllers Example: http://localhost:8080/rest/api For a further breakdown of endpoints see the test script.	1. /api/template 2. /api/loadCrosswalks 3. /api/subject 4. /api/client 5. /api/studentpackage 6. /api/users 7. /api/institution 8. /api/groupofdistricts 9. /api/assessments 10. /api/assessment 11. /api/district 12. /api/crosswalkMap 13. /api/studentGroups 14. /api/user 15. /api/groupofinstitutions 16. /api/groupofstates 17. /api/saveEntity 18. /api/student 19. /api/states 20. /api/entityNames 21. /api/studentGroup 22. /api/state 23. /api/tsbassessment 24. /api/uploadFile 25. /api/proctorassessments
RESTful Response Codes	1. 200 for GET and PUT with response content

	2. 201 for POST with response content
	3. 204 for DELETE no response message
	4. 302 for Redirection (SSO)
	5. 404 for Not supported endpoint
	6. 500 Error(s) from un-supported requests
Describe the Services	**Dependencies**
http	AccommodationService
http	AssessmentService
http	ClientService
http	DistrictService
http	DownloadTemplateService
http	EntityNameService
http	EntityService
http	GroupOfDistrictsService
http	GroupOfInstitutionsService
http	GroupOfStatesService
http	ImportFileService
http	InstitutionService
http	PreviewFileService
http	SaveFileService
http	StateService
http	StudentGroupService
http	StudentService
http	UserService
http	ValidateFileService
http	XwalkService

1. Verification Testing

Verification testing will cover the functional testing of requirements.

Requirement Document	Document Name	Location	Version
Level I Requirements	DRC SBAC11 Requirements – Test Registration Level 1	Knowledge Tree -> High Level Requirements -> DRC	V.2
Level II Requirements	DRC SBAC11 Requirements – Test Registration Level II	Knowledge Tree -> Requirements -> Level II Requirements	V.2

2. Validation Testing

Validation testing will cover scenario testing.

IV. Test Strategy

1. Test Tools
Tools may be used to:
- Invoke business rules by manipulating, creating, deleting or updating test data
- Determine test pass/fail criteria in a manual and automated fashion
- Document requirements to traceability

2. Test Environment
The DRC test environment is located at DRC and supported by the development team. The environment uses the following software, tools and frameworks:

Tool, Framework, Software, etc.	Category	Version
Java	Programming Language	1.7
Spring Framework	Framework	3.2.0
MongoDB	Database	2.0
EC2	AWS	n/a
Oracle VirtualBox	Virtual Machine	4.2.6
Apache Tomcat	Server	7.0.35
JCE	Encryption	
Hyperic	Monitoring Agent	5.0
Rabbit MQ	Messaging Queue	
Jmeter	Testing tool	
Advanced Rest Client Google Chrome	Testing tool	3.1.1
Hexawise	Testing tool	n/a
cURL	Testing tool	7.2.8.1
Chrome, Firefox, IE, Safari	Browsers	
FileZilla	Testing tool – SFTP	3.6.0.2

3. Test Interface
The Test Registration interface provides interfaced web services, search widgets, drag and drop controls and CRUD functionality. Sorting columns within Grids; In addition REST was utilized to perform stateless API tests.

V. Test Deliverables
The table below lists the deliverables providing proof of testing.

Document Name	Type
Test Plan (this document)	PDF
Test Results (this document)	PDF
Test Scripts	PDF
Test Dashboard (this document)	PDF
Test Report (this document)	PDF

Acceptance Criteria

The Test Registration logical component interface is another cog among Assessment Delivery components. The user interface allows for CRUD operations for users, students and others to interact with creating, managing and published assessments for tenants from Test Spec Bank. Consequently, the goal of testing is not to completely demonstrate that currently documented requirements have been exactly met; rather, the goal is to demonstrate that the delivered Test Registration component fulfills the basic purpose of the component and can be enhanced or modified as requirements are discovered and finalized. Test Registration Interacts with Data warehouse, Test Delivery, SSO and Test Authoring, Test Spec Bank and Program Management as well as Monitoring and Alerting.

II.   Introduction

2.   The overall purpose of this component is to manage and maintain assessment data that is needed across the test delivery system and for the Test Spec Bank, Test Packager and other components in the system. This includes:

   a.   A UI to allow for creation and management
   b.   A REST service to return a web-addressable path
   c.   A UI for storing system configuration data
   d.   RESTful services system configuration data and integration data
   e.   A Database of collections of data
   f.   Authentication Integration for login and logout

Users of Test Registration will focus on assessment content registration packages centric for student, student groups and associated data. Clients of the Test Registration component will be system components interfacing with web services for sso, test authoring, test specification bank, data warehouse and test delivery; throughout the delivery approach which may include additional non DRC scoped components.

Terms and Definitions

Term	Definition
Assessment	Data items which may include images to be displayed or included on a user interface. The data items may be associated to a component, and a tenant.
Package	The customization of the appearance of the system for a particular Consortium client. Branding assets include logos and banners.
Core Standards	The portion of Program Management that is responsible for externalizing properties and managing component endpoints.

Content	The portion of Test Authoring responsible for data management
RESTful Endpoint	The location of a service, pointed to by a URI
Program	For the purposes of these requirements, the term "program" refers to a customer's plan for administering tests. A program is not considered to be an entity.
URI	Uniform Resource Identifier, a string of characters used for identifying and locate a resource
Tenancy	Hierarchical structured component
DTD	Document Type Definition provided by AIR
TIB	Test Item Bank
TSB	Test Specification Bank
TR	Test Registration
TAA	Test Administration

Terms and Definitions

Term	Definition
APIP	Accessible Portable Item Profile. A technical standard that focuses on accessibility in assessment items.
Blueprint	The design for a test. The test blueprint indicates the number of test questions or points related to each competency on the test and the relative emphasis placed on each competency.
Field Test	Test made up of test items intended to develop and calibrate new assessments.
Grade Level	Grade or standard the item is designed for
Item	A question on a test. It may be composed of several parts including audio and visual files.
Item Author	A person or agency responsible for item creation.
Item Bank	A systems application that manages the storage and retrieval of assessment items, tracks item versioning and lineage, and provides a robust search and query capability that allows searching on all types of metadata.
Item Status	A term used to indicate how an item can be used. Some of the valid values are: • Pilot Test • Field Test • Operational
JSON	JavaScript Object Notation, a lightweight, text based, open standard designed for human-readable data interchange.
Pilot Test	A trial series of new or modified items given to a select group of students.
Retired Item	An item that will no longer be used for an assessment.
Rubric	a guide listing specific criteria for grading or scoring academic papers, projects, or tests

Runtime exception	An unanticipated condition for which no error handling exists but normal processing cannot continue
Source Item Bank	The Item Bank from which an item is pushed to a Test Item Bank
XML	Extensible Markup Language, a set of rules for encoding documents in machine-readable form.

Data and Data Relationships

The Test Registration user interface allows for creation, management of student, user, assessment, levels information.

File Upload Format	Description
Student	all student attributes, demographics and attributes contained in the Student file upload
Eligibility	all assessment eligibility information contained in the Eligibility file upload

Table 3. Data Warehouse Registration Record Components

Trigger	Description
Update of student information	Any time student information is updated via user interface input or file upload, the Data Warehouse needs to be notified of the change. A full registration record is sent for each assessment the student is eligible for.
Update of eligibility criteria	If eligibility criteria for an assessment is changed such that eligibility changes for a one or more students, a file is downloaded to the Data Warehouse with a record for each student whose eligibility changed (became eligible or ineligible)
Addition of a new assessment	If a new assessment is added and implicit eligibility rules are configured such that some number of students become eligible for that assessment, then a file is downloaded to the Data Warehouse with a record for each student that became eligible for the new assessment as a result of assessing the eligibility rules
Update of explicit eligibility	Any time a student's explicit eligibility for an assessment is updated by means of upload of an Eligibility file, the Data Warehouse must be notified by sending a new eligibility record. Added Infromation: 1. Eligibility rules belong to a particular assessment. 2. Particular assessments have one and only one subject. 3. Eligibility rules are uploaded per subject. 4. Eligibility rules that evaluate accommodations should only evaluate accommodations for subjects that match the subject of the assessment.

Table 4. Data Warehouse Registration Resend Triggers

Role	Description
Administrator	The Administrator is a global role across components and has unfettered access to all component's features
State Coordinator	State Coordinators are able to create other State Coordinator accounts as well as District and School Coordinator accounts and Test Administrator accounts. They are able to upload data across all districts and schools within their states.
District Coordinator	State Coordinators are able to create other District Coordinator accounts as well as School Coordinator and Test Administrator accounts. They are able to upload data across all schools within their districts.
School Coordinator	School are able to create other School Coordinator and Test Administrator accounts. They are able to upload data for their school only.
Test Administrator	Test Administrators can use the Test Registration user interface to search for and view student and accommodation information in a read-only manner.

Table 5. Test Registration User Roles

Requirements

Requirements are numbered according to the following convention:

1. RFP.## - a requirement from the RFP

2. RADPM.## - a requirement from the detailed requirements from RFP-11 or the Architecture document

3. PRPM.## - a requirement from the Proposal

4. DRPM.## - a detailed requirement from the Level I document

5. OTHPM.## - other Program Management requirements

The Test Registration component must meet the applicable requirements from the General Requirements. The following are the example artifacts accompanying this document.

I.

Source.ID	Requirement	Comments
RFP.89	The Test Registration component will provide an interface for querying students that need to be registered for test assessments.	The query response will contain the necessary identifiers to be supplied to the Test Delivery component in order to link students to their appropriate test responses. The UI for this function is part of the Test Administration component.
RFP.89.1	**Input:** SIF compliant XML for student information	
RFP.89.2	**Output:** SIF compliant XML for student information	
RFP.90	The Test Registration component will interface with the Test Delivery component using SIF compliant XML APIs.	
RFP.91	The Test Registration component will provide an interface for accepting student information that is keyed into the Test Administration component	This is for any student(s) that was not uploaded from SIS systems.
RADTIB.1	The Test Registration component will be built with Open Source Technologies The component will contain: • web services, • business logic, • any repositories to implement the registration functionality	
RADTIB.2	The Test Registration component will communicate with student information systems prior to registering students for test sessions	The bulk of the data will be uploaded through SIF compliant APIs.
RADTIB.2.1	**Input:** SIF compliant XML for student information	
RADTIB.2.2	**Output:** Simple XML response meta data (success, error, etc.)	

RADTIB.3	The Test Registration component will supply student identifiers to the Test Delivery component.	
RADTIB.7	The Test Registration component will provide a repository and interfaces for user access	This will include site information that allows the system to determine which users will have access to particular student information and details during the registration process.
RADTIB.8	The Test Registration component will support an interface allowing states to send student information in a non-SIF-compliant format.	The Test Registration component will not transform data to this CSV format.
RADTIB.8.1	The format will be a CSV file created based on the SIF student structure.	
RADTIB.8.2	Non-SIF-compliant data files should be translated to SIF-compliant XML for use with the other components of the overall system.	
RADTIB.9	The Test Registration component will provide performance metrics, error logging, and work flow alerts to the monitoring and alerting component.	This will use the framework specified in the alerts and monitoring component details in a later section of this document.
RADTIB.10	**Student data:** The student data represent the SIF-compliant XML data.	
RADTIB.11	**Status and linking data:** The status and linking data represent the information required to link students to the test sessions.	
RADTIB.12	Test Registration requires knowledge of roles.	Users are assigned one or more roles. Roles, in concert with associated entities, provides components with information they require to grant access to particular component features to users, and to carry out other business rules based on a particular user's assigned roles.
RADTIB.13	Test Registration is responsible for providing entity information to downstream components. The following is the hierarchy of entities from top to bottom: 1. CLIENT 2. GROUPOFSTATES 3. STATE 4. GROUPOFDISTRICTS 5. DISTRICT	

	6. GROUPOFINSTITUTIONS 7. INSTITUTION	
RADTIB.14	User file uploads will contain: • user's name • email • address • phone number • identifier • role • associated entity, the level of the associated entity • entity chain for the associated entity. • grade, demographic information • state • district • institution identifiers	
RADTIB.14.1	There should be one row for each role-entity combination that applies to the user	
RADTIB.14.2	The user's information should be the same from row to row.	
RADTIB.14.3	If the user information varies, the rule is that the "last row wins" rule applies and the user information stored is that of the last row.	
RADTIB.14.4	User file upload includes an Action column that indicates to Test Administration that this is an addition, a deletion or a modification of this user's information.	
RADTIB.15	Accommodation upload files contain the student's • SSID, • the assessment subject for which the accommodations apply • various accommodation codes.	No Action column is required because uploading a student's accommodations either add or update the student's accommodation. The student identified by the SSID in the Accommodation file upload must already exist in the system or a validation error will result.
RADTIB.16	Roster information will identify the roster by: • name, • email address of the user that owns the roster, • state	The Action column indicates whether this student record is to be added to or deleted from the roster by containing "ADD" or "DEL."

	• district • school IDs • SSID	
RADTIB.17	The Eligibility will contain: • SSID • administration name, • subject • name of the assessment • Action column	No update feature is provided; the student eligibility should be deleted from one assessment and added to another.
RADTIB.17.1	User interface in Test Registration will be able to add the following information to the assessment for eligibility:	
RADTIB.17.2	Implicit Eligibility	If implicit eligibility is selected, administrative users must also provide eligibility rules via the Assessment UI in Test Registration.
RADTIB.17.3	Explicit Eligibility	A student is eligible to take an assessment configured for explicit eligibility only if an Eligibility record in an Eligibility file upload explicitly registers the student for the assessment.
RADTIB.17.4	Test Windows	Administrative users must add start and end dates for test windows for particular assessment
RADTIB.17.5	Number of Opportunities	Number of opportunities is entered by the administrative users into the Test Registration user interface for a given assessment.
RADTIB.17.6	Delay Rule	A configurable number of days are provided so that a student must wait at least that many days between test opportunities
RADTIB.18	Download Files to the Data Warehouse	Registration records are sent to the Data Warehouse as soon as Test Registration determines that a

		student is eligible for a particular assessment
RADTIB.18.1	The registration record that is sent to the Data Warehouse consists of concatenation of the same fields contained in the following file upload formats: • Student • Eligibility See Table 3.	
RADTIB.18.2	Test Registration sends one record to the Data Warehouse for each assessment the student is eligible to take. See Table 4	
RADTIB.18.3	Data Warehouse is configured in Program Management to receive de-identified student information, the registration information sent to the Data Warehouse needs to generate with blank fields for: • FirstName, • LastName, • MiddleName • BirthMonth • BirthDay, • BirthYear • SSID • ExternalSSID	There is only one role per row in the import file, but there can be multiple rows. Yes, the UI should permit the assignment of any arbitrary number of roles each associated with an allowable entity. Permissions, who is the owner of roles, also provides an enumeration that indicates the level of the allowable entities to associate with that role. TR should only offer users entities at those levels. For example, if "Test Author" can only be associated with CLIENT and STATE, then TR should only off those and not any other level
RADTIB.18.4	The format of the file downloaded to the Data Warehouse is documented in the Data Warehouse tab in: SBAC 11 Operational System Test Registration File Formats.xlsx.	
RADTIB.19	The Data Warehouse will perform validations on students registration data	One form of validation is to ensure that student demographics are consistent for the same student from year to year. The demographics for a given student may change, but the Data Warehouse may also (for example) check that demographics from multiple students from the same school are consistent from year to year. The results of these validations are transmitted back to Test Registration.

RADTIB.19.1	Provide Identifier Management as the Data Warehouse is allocated the responsibility for tracking students that move from one Smarter Balanced state to another.	The Data Warehouse should provide interfaces for such notifications, and Test Registration provides the user interface for administrative users to enter this information.
RADTIB.20	Use Single Sign-On (SSO) that is provided with user files.	The format of the file downloaded to the Data Warehouse is documented in the SSO tab of 1. SBAC 11 Operational System Test Registration File Formats.xlsx
RADTIB.21	Allow Test Reg to interface with Test Delivery that requires the following information: • Assessment information is required for presentment to the Proctor to select from available assessments to start a test session • Institution information is required to assess business rules that match students to proctors by school • Eligibility information to determine whether students are eligible to take assessments selected by a proctor in a test session • Accommodations information to Test Delivery can provide the appropriate tools and accommodations to the student	
RADTIB.22	Allow Test Reg the Use of Permissions	The Permissions component provides REST interfaces to browse for roles, component permissions and the mapping between these. Components cache this information so that authorization and tenancy can be determined when a user is authenticated.
RADTIB.23	Data synchronization procedures are to be provided by Test Registration for the Data Warehouse and SSO where Test Registration generates master download files tailored for each of these components	
RADTIB.24	Create communication channels between Test Registration and the Data Warehouse	
RADTIB.24.1	Test Registration and the Data Warehouse should communicate over REST interfaces to initiate a file download,	Should provide the means for the Data Warehouse to respond with validation results and completion data when the Data Warehouse has finished consuming

		the download file provided by Test Registration
RADTIB.24.2	Identifier Management: Test Registration should use REST interfaces provided by the Data Warehouse to communicate information regarding a former student registration in another Smarter Balanced state	
RADTIB.24.3	Synchronization: REST interfaces are required to initiate and manage synchronization procedures between Test Registration and the Data Warehouse	
RADTIB.25	Test Registration will provide two REST APIs for Test Delivery to browse assessment data	
RADTIB.25.1	Test Delivery will call this API to retrieve a list of configured assessments when a proctor wishes to open a test session	
RADTIB.25.2	The list of available assessments is presented to the proctor, and the proctor selects from this list to open a test session	
RADTIB.25.3	Authentication, authorization and Information about the proctor is provided by SSO and not by Test Registration prior to calling this API to retrieve the list of assessments	
RADTIB.26	Use the REST API to retrieve a Student Package	
RADTIB.26.1	Test Delivery calls this API when a student provides their credentials to the Student application	
RADTIB.26.2	The API looks up this students information and returns student attributes, demographics, institution chain, accommodations, and information about the assessments the student is eligible to take	This information is used to assess business rules in Test Delivery, and determine which of the assessments on this test session to make available for the student to select
RADTIB.26.3	Interface should be accessed using Single Sign On (SSO)	
RADTIB.26.4	Interface should also validate Permissions	

RADTIB.27	Test Registration provides user access based on five roles: See Table 5	
RADTIB.28	Model features of Test Registration on the TIDE model.	
RADTIB.28.1	TIDE can be used as a model for the following features of Test Registration that are common between various entity types. 1. Landing page and icons displayed 2. Navigation 3. Search functions 4. Search results 5. File upload (specifically the four-step file upload model: choose file, preview file, validate file, confirmation)	Using TIDE as a model is desirable because Smarter Balanced personnel, states, districts and schools are familiar with TIDE features.
RADTIB.29	SB requires that when changes are made to a student file, ADD/EDIT/DELETED that this action be captured and the older file be saved in order for users to audit any changes made to files.	
RADTIB.30	System requires that interface is compliant with Federal 508 requirements with those who have a sight or hearing disability	See both 508 requirements sections below
RADTIB.31	(UI) Create CRUD UI for student group data A UI needs to be created to manage CRUD functionality for Roster data. Also include search, export, and print capability. Testing Information: CRUD only. This does not cover business rules or validations. To test simply do the three things we do (create, update and delete) through UI and verify the information is persisted to the database.	TESTREG-31 has been created

II. User Interface Components

As described above, the Test Spec Bank is a repository of test specifications and does not require a user interface. It only exposes REST APIs to client components.

1. Summary of User Interface Functions

The following table summarizes the user interface features of the Test Registration system and indicates the roles that are applicable to each function.

					Role				
Category	**Function**	**Feature**	**TIDE Manual Reference**	**Description**	**Admin**	**State Coord**	**District Coord**	**School Coord**	**Test Admin**
Assessment	View/Edit Assessment	Search for Assessment	N/A	User searches for assessments; search results are presented in a tabular search results page	⌉				
Assessment	View/Edit Assessment	Delete Assessment	N/A[1]	User selects one or more assessments from tabular search results page and deletes them	⌉				
Assessment	View/Edit Assessment	Export Assessments	N/A[1]	User selects particular assessments from the search results page for export OR chooses to export all assessments in the search results in Excel or CSV format	⌉				
Assessment	View/Edit Assessment	Print Assessments	N/A[1]	User selects particular assessments from the search results page for print OR chooses to print all assessments in the search results	⌉				
Assessment	Upload Assessment	Upload Assessment file	N/A[1]	User uploads an assessment XML package from the Test Packager; this can only add a new assessment or modify an existing	⌉				

				assessment					
Institution	View/Edit Institution	Create Institution	N/A	User creates a state, district or school entity	⌉	⌉	⌉[3]	⌉[3]	
Institution	View/Edit Institution	Search for Institution	N/A[2]	User searches for state, district or school entities; search results are presented in a tabular search results page	⌉	⌉[3]	⌉[3]	⌉[3]	
Institution	View/Edit Institution	Update Institution	N/A[2]	Users selects a particular institution from a tabular search results page and updates it	⌉	⌉[3]	⌉[3]	⌉[3]	
Institution	View/Edit Institution	Delete Institution[3]	N/A[2]	User selects one or more institutions and deletes them	⌉	⌉[3]	⌉[3]	⌉[3]	
Institution	View/Edit Institution	Export Institution	N/A[2]	User selects particular institutions from the search results page for export OR chooses to export all institutions in the search results in Excel or CSV format	⌉	⌉[3]	⌉[3]	⌉[3]	
Institution	View/Edit Institution	Print Institution	N/A[2]	User selects particular institutions from the search results page for print OR chooses to print all institutions in the search results	⌉	⌉[3]	⌉[3]	⌉[3]	
Institution	Upload Institution	Download Institution File Template	N/A[2]			⌉[3]	⌉[3]	⌉[3]	
Institution	Upload	Upload	N/A[2]	User uploads	⌉	⌉[3]	⌉[3]	⌉[3]	

	Institution	Institution File		an institution file containing states, districts and/or schools					
Student	View/Edit Student	Create Student	24	User creates a new student and supplies demographic, institution, accommodation and eligibility information	⌉	⌉³	⌉³	⌉³	
Student	View/Edit Student	Search for Student	17	User searches for students; search results are presented in a tabular search results page	⌉	⌉³	⌉³	⌉³	⌉
Student	View/Edit Student	Update Student	20	User selects a student from the tabular search results page and updates student information including demographic, institution, accommodation and eligibility information	⌉	⌉³	⌉³	⌉³	
Student	View/Edit Student	Delete Student	18	User selects one or more students from the tabular search results page and deletes them	⌉	⌉³	⌉³	⌉³	
Student	View/Edit Student	Update Identifier Management	N/A[1]	User selects a student from the tabular search results page and selects to update	⌉	⌉³	⌉³	⌉³	

				Identifier Management information; this can be a different identifier from the same state or from a different state					
Student	View/Edit Student	Export Student	19	User selects particular students from the search results page for export OR chooses to export all students in the search results in Excel or CSV format	⌉	⌉³	⌉³	⌉³	
Student	View/Edit Student	Print Student	20	User selects particular students from the search results page for print OR chooses to print all students in the search results	⌉	⌉³	⌉³	⌉³	⌉⁴
Student	Upload Students	Download Student File Template	30	User downloads an empty template file to facilitate providing student information in the correct format	⌉	⌉³	⌉³	⌉³	
Student	Upload Students	Upload Student file	30-35	User uploads a student file; this can add, delete or modify students and student and demographic information	⌉	⌉³	⌉³	⌉³	
Student	Upload Accommodation	Download Accommodation	30	User downloads	⌉	⌉³	⌉³	⌉³	

s	s file Template			an empty accommodations template file to facilitate providing accommodation information in the correct format					
Student	Upload Accommodations	Upload Accommodations file	31-35	User uploads an accommodation file that creates student accommodations for particular subjects	)	)³	)³	)³	
Student	Upload Eligibility	Download Eligibility File Template	N/A²	User downloads a student eligibility template file to facilitate providing eligibility information in the correct format	)	)³	)³	)³	
Student	Upload Eligibility	Upload Eligibility file	N/A²	User uploads an eligibility file that makes students eligible for assessments where explicit eligibility is required	)	)³	)³	)³	
User	View/Edit User	Create User	41	User creates a new user and assigns one or more user roles and an associated entity for each user role	)	)³	)³	)³	
User	View/Edit User	Search for User	37	User searches for user; search results are presented in	)	)³	)³	)³	

				a tabular search results page					
User	View/Edit User	Update User	37	Users selects a particular user from a tabular search results page and updates it, including changing user information, adding or deleting roles and associated entities, deleting roles and associated entities, or changing roles/entities or reassigning new entities to existing role	⟩	⟩³	⟩³	⟩³	
User	View/Edit User	Delete User	38	User selects one or more users from the search results page and deletes them	⟩	⟩³	⟩³	⟩³	
User	View/Edit User	Export User	39	User selects particular users from the search results page for export OR chooses to export all users in the search results in Excel or CSV format	⟩	⟩³	⟩³	⟩³	
User	View/Edit User	Print User	37	User selects particular users from the search results page for print OR chooses to print all users in the search	⟩	⟩³	⟩³	⟩³	

				results					
User	Upload Users	Download User File Template	42	User downloads a user template file to facilitate providing user information in the correct format	⌉	⌉³	⌉³	⌉³	
User	Upload Users	Upload User file	42	User uploads a User file that can add, delete or modify users and user information	⌉	⌉³	⌉³	⌉³	
Roster	View/Edit Roster	Create Roster	N/A²	User creates a new roster and assigns an owner and students to the roster	⌉	⌉³	⌉³	⌉³	
Roster	View/Edit Roster	Search Roster	N/A²	User searches for rosters; search results are presented in a tabular search results page	⌉	⌉³	⌉³	⌉³	
Roster	View/Edit Roster	Update Roster	N/A²	User selects a particular roster from a tabular search results page and updates it	⌉	⌉³	⌉³	⌉³	
Roster	View/Edit Roster	Delete Roster	N/A²	User selects one or more rosters from the search results page and deletes them	⌉	⌉³	⌉³	⌉³	
Roster	View/Edit Roster	Export Roster	N/A²	User selects particular rosters from the search results page for export OR chooses to export all rosters in the search results in	⌉	⌉³	⌉³	⌉³	

				Excel or CSV format					
Roster	View/Edit Roster	Print Roster	N/A[2]	User selects particular rosters from the search results page for print OR chooses to print all rosters in the search results	⎱	⎱[3]	⎱[3]	⎱[3]	
Roster	Upload Roster	Download Roster File Template	N/A[2]	User downloads a roster template file to facilitate providing roster information in the correct format	⎱	⎱[3]	⎱[3]	⎱[3]	
Roster	Upload Roster	Upload Roster file	N/A[2]	User uploads a Roster file that can add or delete students from a roster	⎱	⎱[3]	⎱[3]	⎱[3]	
Synchronization	Download Synchronization Data	Download Synchronization data to Data Warehouse	N/A[2]	Downloads all available registration records to the Data Warehouse	⎱	⎱[3]	⎱[3]	⎱[3]	
Synchronization	Download Synchronization Data	Download Synchronization data to SSO	N/A[2]	Downloads all available User records to SSO	⎱	⎱[3]	⎱[3]	⎱[3]	

Table 5. Summary of User Interface Functions

VI. Test Dashboard

	A	B	C	D	E	F	G	H	I	J
1	Component	Test No.	Test Case Scenarios			Tests Fulfilled		Build	Expected Delivery	
2	SB-11 Test Registration					Pass	Fail	v0.0.1	May 6,2014	
3		Script 1	API			1	0			
4		Script 2	Create /Modify Students			1	0			
5		Script 3	Create /Modify Users			1	0			
6		Script 4	Delete Users			1	0			
7		Script 5	Upload File, File data			1	0			
8		Script 6	Download Templates			1	0			
9		Script 7	Cross Walk & Load Crosswalks			1	0			
10		Script 8	SSO,Export, datawarehouse			1	0			
11		Script 9	Eligibility Assessments			1	0			
12		Script 10	Eligibility RDI			1	0			
13		Script 11	Eligibility NRDI			1	0			
14		Script 12	Test Auth UI General			1	0			
15		Script 13	States			1	0			
16		Script 14	District, Group of District			1	0			
17		Script 15	System Users			1	0			
18		Script 16	Validations			1	0			

VII. Executive Test Summary

Our primary test objective was to prove Test Registration component was capable to execute level I and level II requirements and received for required functional areas which are covered in the test plan. Component integration testing performed among Data Warehouse, SSO Export, Test Authoring, Test Spec Bank, Test Delivery endpoints and in conjunction with Core Standards web services and SAML / SSO authentication across endpoints.

Component interactions, tenancy and deployment tests cover URI, REST operations, RESTful web API HTTP controllers, layers, JSON responses, database collections, UI/persistence, database and web services, authentication, roles and permissions.

Test coverage consisted of positive, negative, boundary and user scenarios based from L2 requirements for assessment package of REGISTRATION, student, users, SSO, tenancy levels, inheritance of configurations, program management dependent functionality, filtering, sorting, workflow and user interface navigations.

Browser performance and compatibility UI testing was performed cross browsers using Chrome, Fire Fox, Safari and IE where default settings were tested. http: tests covered look and feel standards, services, functionality, logic and usability.

UI testing was based from supporting UI AIR guidelines and direction from their UI team experts that provided unique icons for functional areas, navigational pages and designs and layouts for Test Registration user interfaces.

Test Equipment / Tools – include but are not limited to: a repeatable infrastructure environment and intel core i5 laptops accessing the /api and using either Fire Fox, IE,Safari or Chrome browsers and their respective REST client plugins. Mac book pro with retina quad core i7 and a linux Ubuntu 12.04

All defect discoveries or improvements revealed during testing were quickly reported, managed and fixed in JIRA. Test coverage analysis compiles from combinatorial pairs, cobertura reports and requirements.

The heuristic test strategy, a simple model was used considering the requirements, tests, and environment and product elements.

Overall the complexities of Test Registration component are functional and integrated among existing delivered components as well as being load tested with Data Warehouse for over 5million students; as well as being tested for proctor and student endpoints to Test Delivery. Test Registration shows to have good design, reliability, scalability, usability, and capability as a supporting features component within a shared services environment where expected future component dependencies will rely on Test Registration to fulfill service for authorized user communication of information throughout component logical interfaces.