System Test Specification

Hayden Eubanks

October 8, 2023

Table of Contents

1. Test Cases

Test Case ID	Requirement Number	Test Goal
LOGIN-1	1.1	**Validation of user Passwords:** The validation of user passwords can be tested by creating test cases including all forms of invalid password data entries and then passing those test cases to the system to ensure they are not accepted. This could include factors such as password length, the inclusion of numbers and special characters, and not solely using words from the dictionary. Further, as this is an input field, testing should be performed to ensure special characters used in injection attacks cannot be injected into the system.

LOGIN-2	1.1.1	**Minimum Password Requirements Met:** This testing area is an extension of the first test case regarding password requirements but extends this test by testing for the security of password elements as a whole unit. For example, a password must contain at least four-letter characters, one special character, and one number to make a minimum password of six characters in length and contain a combination of elements that increase the password's security. Further, poor security practices with these elements such as commonly used patterns (i.e., “abc123!”) or the repetition of characters (i.e., “111aaa!!!”) will be tested for and passwords that practice these poor standards will not be accepted.

LOGIN-3	1.1.2	**Password Hashed and Hash Value Stored for Authentication:** The hashing of passwords and their associated stored values can easily be tested by inspecting the stored values in the database to ensure that fixed hash values are implemented. Further, testing of using the same password for different user accounts could be implemented to ensure these values hash to different outputs indicating the existence of salt in hashing. However, from a purely black-box testing perspective, the data traffic could be inspected to ensure that the password value is encrypted by the hashing algorithm before it is sent to the database for storage.
LOGIN-4	1.1.3	**Passwords Must Regularly Be Updated:** This test case can be performed by setting a password and ensuring the password expires after the given time,

		being three months in this system, and forces the generation of a new password. However, with access to the system code an inspection of the expiry value of passwords can be inspected to ensure the facilities for password timeout exist. Further, the system should remain inaccessible to the user until the password is reset.
LOGIN-5	1.2	**User Sessions will Timeout After a Period of Inactivity:** This test feature can be tested by initiating a user session and then ensuring the system logs the user out and terminates the user session after five minutes of inactivity. Creating a user session and manually verifying the system status after five minutes can then confirm this metric. Additionally, the system should not sign out if the user is still utilizing the system so a warning should be given to the user and the system

		should not initiate the auto logout if actions are still being performed.
LOGIN-6	1.3	**Only Connections from Within the U.S. Should Be Accepted:** This can be tested by system testers through the utilization of VPNs to imitate connections from other nations. In all cases except for connections from the U.S., the system should reject the connection and terminate the session without allowing access to system resources. In contrast to this, all verified connections originating in the U.S. should be accepted and access allowed.
PERFORM-1	2.1	**Efficient Querying of Database:** The database should return queried values within an acceptable timeframe. The general rule for testing this

		metric would seek to return query results in under five seconds on average. This can be tested by generating SQL queries of varying complexity and performing these queries on the database while tracking the time until data is returned. The average time can then be taken over the course of all tests to ensure the system performance is within an acceptable range.
PERFORM-2	2.2	**Sufficient Resources for College User Base:** This testing could best be performed through a physical inspection of the storage and computational resources available to the server relevant to the number of students needing to be served by the system. This would most easily be accomplished through inspecting the system infrastructure and resources but could also be accomplished through simulating the expected load of traffic on the system

		and ensuring all systems continue to function normally.
PERFORM-3	2.3	**Sufficient Data Throughput for Connections:** As an extension of the previous test case, this test involves simulating user activity on the server specifically and ensuring that the data throughput returns data to the user within an acceptable timeframe. This can be tested by simulating several user connections under the expected load and ensuring a timely response for data retrieval is present.
PERFORM-4	2.3.1	**TCP Implemented for User Connections:** This test case can easily be performed by creating user connections to the server and inspecting the protocols used in data transfer. A tester should then observe that the TCP protocol is utilized for all transfers of data.

PERFORM-5	2.4	**Critical Sections of Data Modification Protected:** Test cases can be crafted to examine the functionality of data-critical sections by testing scenarios in which race conditions could occur. For example, a user could attempt to modify a piece of data while another is viewing that data, or two users could attempt to simultaneously modify the same data element. In all of these examples, the integrity of data should be upheld through the protection of critical data sections.
DATA_PROTEC-1	3.1	**Encryption of Data at Rest:** The encryption of data at rest would almost exclusively be testable by inspecting the database values and ensuring that encryption is implemented for the data values. A tester could then examine server files to ensure encryption practices are implemented across all stored data files.

DATA_PROTEC-2	3.2	**Encryption of Data in Transit:** The encryption of data in transit is more easily testable as the values of data in transit can be inspected through an application for inspecting network traffic. Inspecting network traffic should then reveal no cleartext transfers implying that encryption is practiced for all transferred data values.
DATA_PROTEC-3	3.3	**Principle of Least Privilege Applied Throughout Design:** The principle of least privilege should be practiced throughout the design and can only be tested with a thorough understanding of the roles associated with each user class. With this understanding, a tester could then attempt actions as a user of each user group and ensure that only needed actions are performable with each user group type.

DATA_PROTEC-4	3.4	**Audit Log Maintained of all Data Transformations:** In a similar manner to the other test cases revolving around data stored on the server, the testing of audit log values can only be performed by inspecting the audit log and performing actions that should be recorded on the system. The audit log should then reflect these actions and be observable by the system tester.
DATA_QUAL-1	4.1	**All Mandatory Fields Must Contain a Value:** This test case can be tested by examining the outcome of leaving each mandatory field empty. In the end system, this should result in the form not being submitted and a warning message displayed to inform the user that they must add data to the input field.

DATA_QUAL-2	4.2	**Every Student Has a Data Entry:** Similar to the previous test case, this test case can be implemented by scanning the database against a known list of students attending the university. For this test to be successful, it must then be observed that every student represented in the list holds an entry in the database. This test could also be used to identify errors in student names and all misrepresentations must be manually inspected.
DATA_QUAL-3	4.3	**Data Validation is Performed on All Fields:** This test case can ensure data quality by testing every input field for the adequate validation needed to ensure high-quality data from that field. This could look like generating testing parameters concerning areas such as the characters allowed in a field, the length of the field, and the correct data type are validated for each

		entry field. All fields must be tested but test cases can be written and automated to streamline this testing process.
DATA_QUAL-4	4.4	**User Records Deleted After Allotted Time:** A lifespan should be generated for all data stored within the system to ensure all data kept is relevant to the college. This can be tested by creating a new data entry and ensuring that this data entry is removed from the system after the allotted time. However, this black box testing approach may not be desirable as the expiry of data will occur after five years. For this reason, artificial aging should be performed on data elements to ensure that this feature works correctly when the age of data is set past the expiry value.

LEGAL-1	5.1	**FERPA Regulation Compliance Throughout Design:** This compliance can be tested by comparing the current implementation against the policy and guidelines outlined in the FERPA regulation. This area reflects the testing of policy, and for this reason, can be tested by examining the current policy involved in the implementation process and then ensuring that adequate technical controls are expressed in the design.
LEGAL-2	5.2	**All U.S. PII Best Practices and Legislation Must Be Followed:** In a similar manner, this test case can be performed by generating automated tests that explore the compliance of the system's implementation to ensure that all regulation is followed in protecting personally identifiable information. These test cases will be quite

		specific as they will reflect specific guidelines in the regulations and as such will clearly distinguish if the system is in compliance.
SCALABILITY-1	6.1	**System Must Be Easily Scalable to Accommodate Student Growth:** This factor can be easily tested by examining the current usage data of the system and then calculating the predicted usage statistics regarding the expected growth over the coming years. The system can be seen to adequately accommodate for growth if the newly predicted values are still within the range of acceptable performance.
SCALABILITY-2	6.2	**The system Must Be of Adequate Size to Hold All Student Records Plus Growth Buffer:** Extending from the previous test case, this test case involves examining the storage capacity utilized by the

		current database of students and then examining the remaining storage space to ensure that adequate storage for growth is present throughout the system. This can be accomplished by calculating the average storage capacity of a single database entry and then calculating the expected future storage needed.
SCALABILITY-3	6.3	**System Brought Up to Compliance for International Use:** While not an immediate requirement for the system, this test case can be performed by examining the system's implementation to highlight areas that would stop the system from being compliant with international communication standards so a list of features to change is readily available when international communications are desired to be added.

SYS_PROTEC-1	7.1	**Students Must Be Able to Request Corrections to Their Data:** This test case can easily be carried out by creating a student user account and then ensuring all data fields can be requested for change. This could involve manually attempting to request a change of each data field present in the student database entry.
SYS_PROTEC-2	7.1.1	**Only the Data Owner Can Request Data Changes:** Like the test case before it, this test case can also be carried out by creating a student user and then attempting to request the data change of another student user's data. The system should not allow this request to be processed, but even further should not allow a student access to another's data entry in the database.

SYS_PROTEC-3	7.2	**Acceptable Use Policy Implemented:** The acceptable use policy can be used for this test case to examine the adequate enforcement of the policy throughout the system. This test case primarily regards the testing of policy and as such could further be tested through the performance of workplace audits regarding compliance with the policy.
SYS_PROTEC-4	7.3	**DOS Protection:** This test case could be performed through penetration testing methods where a mock DOS attack could be performed against a copy of the system to ensure the system adequately defends against the DOS attack. The system's availability should not suffer as a result of the attack to ensure that this requirement is fully met.

SYS_PROTEC-5	7.3.1	**TCP Connections Will Time Out After Inactivity:** This can be tested by generating TCP connections and then observing them to ensure the connections are terminated after the desired period of inactivity. In this implementation, the inactivity period has been set to fifteen minutes and the test cases should reflect this value.
SYS_PROTEC-6	7.4	**User Action Verification:** This test case revolves around the requirement that all user actions must verify that the user attempting them has the appropriate privileges. This can be tested by generating a user of each type and then attempting to perform every action possible with that user type. For this test to be successful, only the actions associated with that user group should be allowed with the rest

		being rejected and an error message displayed indicating the action is not allowed for that user.

2. Test Specification

2.1. LOGIN-1 - Validation of user passwords

Steps	Sub-Steps
Check Password Length	Ensure the Password contains at least six characters
	Ensure the Password contains at most fifteen characters
Check for Special Characters	Encode all special characters
	Ensure at least one special character is used
	Ensure that no banned special characters are used
Check for Numbers	Ensure the Password contains at least one number

Accept or Deny the Password	If the Password is compliant with all tests, accept the password and process the request
	If any tests fail, deny the password and display the error message

Test Completion Indicator: "Password meets all desired criteria" – Password validation is accepted, and the password is then hashed and stored in the user database.

Evaluation Process: Inspect passwords accepted by the system to manually validate that all requirements are complied with in accepted passwords.

2.2. LOGIN-2 - Minimum password requirements met

Steps	Sub-Steps
Ensure No Common Words are Used in Isolation	Simple dictionary words should be rejected when not combined with other words or characters

	Ensure dictionary words used are at least five characters long
Ensure Minimum Character Values	Ensure the password is at least six characters long
	Ensure the password contains at least one number
	Ensure the password contains at least one special character
Ensure Character Variation	Ensure at least three different alphabetic characters are used
	Ensure that at least six different characters are used overall
Deny Commonly Used Passwords	Test given password against a list of most common and vulnerable passwords
	If a vulnerability is identified, deny the password

Test Completion Indicator: “Password meets all minimum security requirements” – Password is accepted and updated as the user’s password.

Evaluation Process: Passwords accepted into the system can be examined to ensure that all compliant passwords are accepted while non-compliant passwords are rejected.

2.3. LOGIN-3 - Password Hashed and hashed values stored for authentication

Steps	Sub-Steps
Ensure password values are hashed	Create a new acceptable password
	Save the password to the user account
	Inspect network traffic to ensure encrypted value is transferred to the database

Ensure hashed values are present in the database	Inspect the database to ensure password value is hashed and stored in its encrypted form
Ensure the same password hashed by different users results in different hash values	Repeat the first two steps of creating a password with the identical value to the first step Inspect hashed value in the database Ensure the hashed value differs from the first hashed value indicating the presence of salted hashed values for added security

Test Completion Indicator: "Passwords successfully hashed and stored in database with hash values being salted for added security" – Each password's hashed values are then stored in the database for comparison upon user login to the system

Evaluation Process: This process can be evaluated by inspecting the database and ensuring that all input values are hashed and further repeated input values output varying hashed values.

2.4. LOGIN-4 - Password must be regularly updated

Steps	Sub-Steps
New Password Generated	A new password is created and accepted by the system The system assigns an expiry value to the password
Password aged	Password artificially aged by changing password age value to after the expiry length
Password Aging Validated or Denied	Password status inspected Password should be denied, and new password creation forced

Test Completion Indicator: "Password aged successfully, and user password resetting forced" – The user is forced to create a new password and denied access to system resources until this action is taken.

Evaluation Process: Attempt password login with an aged password to ensure that access to system resources is denied.

2.5. LOGIN-5 - User Sessions will timeout after a period of inactivity

Steps	Sub-Steps
Create a new user session	Login as a user initiating a new user session with an initialized session age of zero
Age Session	Artificially increase the age of the session to a value greater than the expiry value
Ensure the Session is Terminated	Attempt to further access the session
	Ensure that access is denied to system resources and re-login enforced

Test Completion Indicator: "Session has timed out successfully" – The system successfully times out and with this, the session is terminated denying user access until the user is reauthenticated during a new session.

Evaluation Process: This process can be evaluated by ensuring that an aged session is not allowed access to system resources and likewise that premature denial of access is not performed.

2.6. LOGIN-6 - Only connections from within the U.S. should be accepted

Steps	Sub-Steps
Test a connection within the U.S.	A connection from within the U.S. should be attempted It should then be verified that the system allows the connection to proceed and function normally
Log in to a VPN	A VPN session should then be initiated to allow for the testing of international connections
Test International Connection	An international connection can then be tested on the system

	The system should deny the connection and disallow access to any system resources

Test Completion Indicator: "International connections successfully denied" – When the system can be seen to only allow connections from within the U.S., the test case can be said to have been completed successfully.

Evaluation Process: This process can be evaluated by inspecting connections under varying contexts to ensure that only U.S. connections are accepted.

2.7. PERFORM-1 - Efficient querying of database

Steps	Sub-Steps
Generate SQL queries to represent varying query types	Generate SQL queries of varying complexity
	Ensure SQL queries attempt to access every database file

	Ensure queries are generated for each of the two databases
Time query response to evaluate effectiveness	Perform each query while timing the return time of results
	Confirm that each result returns within the acceptably defined timeframe

Test Completion Indicator: "All queries performed within acceptable timeframe" – The test can be verified as complete when all query tests return within the desired timeframe criteria.

Evaluation Process: This process can be evaluated by generating complex database queries and ensuring the results are returned within the acceptably defined timeframe values.

2.8. PERFORM-2 - Sufficient resources for college user base

Steps	Sub-Steps
Examine current resource usage	Examine the current load on system resources
Determine needed resources	Compare this usage to the number of students the system serves
	Calculate per/user resource consumption
Confirm sufficient resources exist	Examine current system resources to ensure sufficient resources exist to meet the calculated needed value.

Test Completion Indicator: "Sufficient resources present to fulfill system objectives" – When the amount of resources available to the system is sufficient for the system's use case, the test is said to be complete.

Evaluation Process: This process can be evaluated by checking the calculations and ensuring the calculation process is sufficient for accurately estimating the system's usage.

2.9. PERFORM-3 - Sufficient Data throughput for connections

Steps	Sub-Steps
Simulate system usage	Simulate usage on the system within a controlled context to get a baseline for data throughput
Measure throughput levels	Measure the throughput levels given this data load
	Determine the per-user effect on data throughput levels
Confirm the sufficiency of the throughput values	Confirm that the throughput levels are sufficient to sustain the expected system userbase

Test Completion Indicator: “Throughput is sufficient to sustain system activity” – If the throughput potential exceeds the expected values needed, then the system test is complete.

Evaluation Process: This process can be evaluated by comparing the expected throughput needed to the system's capabilities to ensure sufficient throughput can be provided.

2.10. PERFORM-4 - TCP implemented for user connections

Steps	Sub-Steps
Attempt a connection	Attempt user connections given all contexts in which a user may issue a request to the college’s system
Inspect network traffic to ensure a TCP connection has been initiated	Using a network inspection tool, inspect the network traffic to observe network communications.

	Confirm that in all instances, a TCP connection is established.

Test Completion Indicator: "TCP connections established for all user connections" – Once every form of user connection has been tested, the test can be labeled complete, and the results returned for confirmation or denial of the fulfillment of test objectives.

Evaluation Process: Evaluation can be performed on this process by ensuring that all connections establish a TCP connection and further that no other communication protocols are attempted.

2.11. PERFORM-5 - Critical sections of data modification protected

Steps	Sub-Steps
Develop a list of all race condition scenarios	

	List race conditions involving editing while a user is viewing data
	List race conditions that involve two users editing data at the same time
	List race conditions that involve a user attempting to view data that is being modified
Test All conditions	Test each of the race conditions generated in the previous step
Repeat tests many times and observe for anomalies	Repeatedly perform these tests to generate confidence that race conditions do not ensue on the system
	Observe for output indicative of a race condition such as the viewing of old data that has since been modified by the system

Test Completion Indicator: "Critical sections sufficiently protected within the system." – The test can be deemed complete when the generated test results reveal no anomalies in test-case data.

Evaluation Process: This process can be evaluated through crafting scenarios intentionally designed to generate race conditions and then performing those tests to ensure a race condition does not ensue.

2.12. DATA_PROTEC-1 - Encryption of data at rest

Steps	Sub-Steps
Create data to be passed to the database	Create data to be passed to every input field found throughout the system
	Input values into each data field
Access the database to inspect the stored value	Access the database and inspect the database file

	Ensure value is encrypted and not decipherable
	Repeat tests for all files found within the database

Test Completion Indicator: "All stored data successfully encrypted" – The test can be labeled as complete when it is determined that all input fields generate an encrypted value to be stored in the database.

Evaluation Process: This process can be evaluated by inspecting all of the database files to ensure no values are present in their cleartext form.

2.13. DATA_PROTEC-2 - Encryption of data in transit

Steps	Sub-Steps
Generate values to be passed to each user entry field	Create data to be passed to each entry field

	Enter data into each entry field to be passed to the database
Inspect network traffic for encryption	Inspect network traffic Ensure all values being passed to the database are encrypted so that a malicious observer cannot decipher them.
Generate SQL requests for database	Generate SQL requests of various types to the database
Ensure values are encrypted	Inspect network traffic again to ensure that values transferred from the database to the user are encrypted and not decipherable by a malicious observer

Test Completion Indicator: "All input values encrypted in transit" – This test can be validated as complete when all transported values are observed as encrypted.

Evaluation Process: Network traffic can be inspected to ensure that no values are transported in their cleartext form.

2.14. DATA_PROTEC-3 - Principle of least privilege applied throughout the design

Steps	Sub-Steps
Create a user for each group type	Create a new user of each user group for testing user actions
	Make sure an understanding is held of the actions that each user group should be able to perform and not perform.
Attempt all tasks as each user type	Using each user class, attempt all actions on the system including actions that should not be allowed for that user type

Ensure no action is performed that is irrelevant to a user group	Ensure that the user is allowed to access all functions they are permitted to access
	Ensure the user is not able to perform any tasks they are not permitted to perform

Test Completion Indicator: "All user groups exhibit desired access control enforcing the principle of least privilege" – After all actions have been attempted with all user classes, the system can then be confirmed to have accomplished the implementation of the least privilege principle.

Evaluation Process: This process can be evaluated by creating a user of each user class and attempting to access the actions available and not available to that user.

2.15. DATA_PROTEC-4 - Audit log maintained of all data transformations

Steps	Sub-Steps
Create a list of actions resulting in data transformation	Think of all actions involving data transformation. List these actions and generate tests to modify data in each of these instances
Perform actions resulting in data transformation	Perform all actions listed in the first step to ideally generate the log requests
Observe audit logs after data transformations have occurred	Open the audit logs and inspect for all of the data modification actions performed on the database.
Ensure all data transformations are present in the logs	Ensure that each data modification is reflected in a log entry with sufficient detail to identify the modification and the user who performed the modification.

Test Completion Indicator: "All data modifications present in system logs" – This test case can be labeled as complete when every data modification action can be seen to generate a log entry.

Evaluation Process: This process can be evaluated by opening and inspecting the audit logs for the system.

2.16. DATA_QUAL-1 - All mandatory fields contain a value

Steps	Sub-Steps
Open a form to create a new user entry	Operate under the assumption that a new student entry is to be added to the database
	Open form where user data is added to the database
Test each field to see how it responds to lacking input	Starting with the first field but filling in all others, work down the list testing the state of the form with the field being empty.

	Continue to the next field and test for lacking data, noting if the field is mandatory or optional
Confirm that the form will not be submitted unless all mandatory fields contain input	Confirm that all mandatory fields require data for the form to be submitted
	Confirm that all optional fields do not require data for the form to be submitted.

Test Completion Indicator: "All mandatory fields must contain data for the form to be submitted." – The test will be completed when all fields in the form have been tested for their reaction to not holding data.

Evaluation Process: Through testing each field individually, the form can be validated to ensure that all mandatory fields require data for form submission.

2.17. DATA_QUAL-2 - Every student has a data entry

Steps	Sub-Steps
Get the list of all students who should be in the database	Using the hard copies of student records that currently exist, create a list of all students whose data should be present in the database
	Create this list in a form that can easily be compared against the database values
Access the database	Access the database files
	Create a list of all student entries currently present in the database
Inspect the database to ensure each student in the initial list is present in the database	Compare the list of students in the database against the list of students that should be in the database.
	Confirm that all students who should hold data in the database do have database entries.

Test Completion Indicator: "All students have entries in the database" – The test will be marked as complete when all student entries can be observed within the database.

Evaluation Process: This process can be evaluated by opening the database files and ensuring that all students have entries in the database.

2.18. DATA_QUAL-3 - Data validation is performed on all fields

Steps	Sub-Steps
Identify all data entry fields	Scan through the system to identify all data entry fields
	Compile a list of these fields for future reference

Test field for data validation	Ensure each field encodes special characters. Ensure each field accepts only the type of data required Ensure that each field complies with the desired field length
Confirm that all fields consistently mandate data validation	Confirm that all validation parameters are met for each entry field in the generated list of fields

Test Completion Indicator: "All fields correctly validate data entry" – This test case can be verified as complete when all fields generated in the initial list have been tested for validation parameters.

Evaluation Process: This process can be validated by selecting a random assortment of data entry fields and ensuring validation is performed on each of the fields.

2.19. DATA_QUAL-4 - User records deleted after allotted time

Steps	Sub-Steps
Create a new user record	Create a new user record that will be added to the database
	Ensure that the age value for this record is initialized correctly
Age user record beyond the expiry date	Set the age for this record to a value beyond the expiry date for which records should be deleted
Inspect the state of the user record	Inspect the database to ensure that the record is removed from the database

Test Completion Indicator: "Records are removed from the database when the age limit is reached" – This test can be confirmed as complete when the aging process is performed on database records and the records can be observed as appropriately removed from the database.

Evaluation Process: This process can be validated by attempting to create new database record entries with ages past the expiry date to ensure they are removed from the database correctly.

2.20. LEGAL-1 - FERPA regulation compliance throughout design

Steps	Sub-Steps
Generate a list of FERPA regulation guidelines	Read through the FERPA regulation and gain an understanding of the expectations facing the organization. From this regulation create a list of guidelines that must be implemented through both policy and technically in the software implementation.
Ensure guidelines are implemented through policy	For each guideline listed, ensure a policy is enforced to express that guideline in the implementation of the software project.

Ensure guidelines are implemented through technical controls	For each guideline identified, ensure a technical control is implemented to enforce the guideline in the technical implementation of the software.

Test Completion Indicator: "FERPA compliance is enforced throughout the software's design." – The test case can be seen as complete when each FERPA guideline requirement is investigated in both the policy and technical implementation of the software.

Evaluation Process: Inspecting existing policies and implementations regarding this regulation can assist in identifying if the requirement is met in the final product.

2.21. LEGAL-2 - All U.S. PII best practices and legislation must be followed

Steps	Sub-Steps
Identify key areas of the software product that	

involve personally identifiable information	Search the software to identify all areas that engage with PII Inspect the database where PII is stored and accessed
Test each of the identified areas for compliance with existing legislation and best practices.	For each area that engages with PII a thorough inspection of PII regulation in reference to that specific area should manually be performed A decision regarding compliance in that area should then be made and recorded A decision should then be made to confirm or deny PII compliance of the system as a whole

Test Completion Indicator: "Compliance to PII regulation is practiced throughout the entirety of the system" – The test can be marked as complete when each area that involves PII is investigated and verified as compliant.

Evaluation Process: This process can be evaluated through the manual inspection of the system in regard to the system's compliance with PII regulations.

2.22. SCALABILITY-1 - System must be easily scalable to accommodate student growth

Steps	Sub-Steps
Calculate current usage statistics	The current usage statistics can be calculated by examining the system under the current usage load
	From this current usage, a set of comparison statistics can be calculated for the system

	These statistics can be included in a file for comparison with future statistics
Calculate expected growth statistics	With the per-user statistics generated, the same statistics can be calculated for a future predicted student number
	These statistics can additionally be put into a file for comparison with the current statistics
Ensure resources exist to accommodate system growth	The two files of current and future resources can be compared
	From this comparison, an evaluation can be performed to identify if adequate system resources exist to accommodate growth within the system

Test Completion Indicator: "Sufficient resources exist to accommodate expected growth." – The test can be marked as complete when an evaluation of the sufficiency of the system resources is produced.

Evaluation Process: This process can be evaluated by performing additional metric checks regarding expected growth and comparing these results against the results generated from this test.

2.23. SCALABILITY-2 - The system must be of adequate size to hold all student records plus growth buffer

Steps	Sub-Steps
Determine Currently used system storage	Identify the current storage utilized by the system data
	From this statistic calculate the per-user effect on the overall storage utilization
	Calculate the current number of students that the system can support
Calculate expected student growth	Calculate the expected student growth over the coming years that this system may be implemented

	Compare the storage capacity needed for this number of students with the current capacity limits
Make decisions as to the current system's sufficiency	If the system is not currently adequate to support the expected growth, this test will be denoted as failed Otherwise, if the system does have adequate resources, the system is marked as a success

Test Completion Indicator: "Current storage capacity is sufficient to accommodate student growth" – This test will be marked as complete when a determination is made regarding the ability of the system to accommodate the calculated storage needed for student growth.

Evaluation Process: This process can be evaluated by generating a server copy and simulating storage elements on this server to verify expected storage values.

2.24. SCALABILITY-3 - System brought up to compliance for international use

Steps	Sub-Steps
Examine international compliance factors	Examine all regulations and international standards that must be followed to achieve success in international communication
	Generate a list of these factors for testing against the system
Test system against each of these factors	For each factor, test the relevant system components to evaluate current readiness for international communications
	Generate a list of all areas in which the system is found lacking regarding international compliance
	As this is for future implementation, return this list and generate action plans to bring the system into compliance

Test Completion Indicator: "List of improvements needed for international communication generated" – This test can be marked as complete when the list of improvements needed is generated and returned to the user.

Evaluation Process: This process can be evaluated through the manual inspection of current international compliance and comparing the manual report against the automated one.

2.25. SYS_PROTEC-1 - Students must be able to request corrections to their data

Steps	Sub-Steps
Generate a new student user with incorrect data values	Create a new student user with errors in each data field
	View these data fields in the newly created student user profile
Attempt to Request a change for each field	Request modification for each field belonging to the user

	Record any fields where the request fails to generate
Ensure each field allows the user to submit a request for changing the value of that field	Ensure each field allows for the submission of the request for field modification
	Continue testing all fields until every field has been tested

Test Completion Indicator: "Student is able to request modification for all data fields" – This test can be marked as complete when it can be verified that a student user is able to request data modification for each field of input.

Evaluation Process: This process can be evaluated by opening an additional user profile and verifying that a request is successfully generated for data field modification.

2.26. SYS_PROTEC-2 - Only the data owner can request data changes

Steps	Sub-Steps
Generate a student user	Create a new student user account for testing
Attempt to access the resources of another user	Attempt to access the data of another student
	Ensure that the student is unable to access data outside of their own
Attempt to request data modification for a user who is not the active user	Assuming the student somehow gains access to another's data, attempt to request a data modification without the student being the owner of the data.
	Ensure that the request is denied and that the data modification request is not generated
	Ensure the request is logged in the audit logs for inspection

Test Completion Indicator: “System verified to only let student data owner’s request modification” – This test can be seen as complete when the thorough analysis of a user’s ability to access another’s data returns negative.

Evaluation Process: This process can be evaluated by attempting to access another student’s resources from a student account to ensure that only the data owner can access their own data.

2.27. SYS_PROTEC-3 - Acceptable use policy implemented

Steps	Sub-Steps
Create a list of the requirements outlined in the acceptable use policy	Analyze acceptable use policy for compliance guidelines
	Generate a list of requirements that must be present in policy and the system
Ensure the policy is enforced within the workplace and other areas of access	Compare these requirements against the current policy enforced
	Ensure the existing policy is sufficient to enforce the acceptable use policy in the design
Ensure the acceptable use policy is implemented through technical controls in the software	Compare these requirements against the current technical software implementation
	Ensure that the technical controls exist to monitor and subvert noncompliance to the acceptable use policy

Test Completion Indicator: "Acceptable use policy upheld through current implementation" – This test case can be seen as completed when the evaluation of compliance throughout the entire system has been performed.

Evaluation Process: This process can be evaluated through an inspection of policy enforcement as well as verifying that adequate technical controls are implemented within the system.

2.28. SYS_PROTEC-4 - DOS protection

Steps	Sub-Steps
Create a copy of the server for testing against	Create a virtual instance of the server with identical resource capabilities
	Generate a point from which the mock DOS attack will be performed.
Automate the continuous opening of partial TCP	

connections to attempt to drain system resources	Continuously generate partial TCP connections that in theory should clog the system and remove legitimate access to system resources
	Continue this process to the greatest possible capacity or until the system fails
Confirm that system resources remain available	Verify that the system resources remain available
	Ensure that the DOS attack was detected and that the proper defense mechanisms engaged

Test Completion Indicator: "System resources remain available when DOS attack is attempted" – This test can be seen as complete when the DOS attack has reached its full capabilities and in light of this the system remains accessible.

Evaluation Process: This process can be evaluated by performing stress testing on the actual server to determine sustainable traffic limits the system can tolerate as well as the system's ability to detect DOS attacks and defend from them accordingly.

2.29. SYS_PROTEC-5 - TCP connections will time out after inactivity

Steps	Sub-Steps
Create a TCP connection for a new user	Generate a new TCP connection between the user and the server
Let TCP connection age past the time limit	Allow the connection to remain open past the expiry limit of fifteen minutes with no activity
Ensure the connection is terminated	Ensure the connection does not terminate before the fifteen minutes has elapsed
	Ensure that the connection is terminated upon reaching fifteen minutes of inactivity
	Ensure that further activity before the expiry time is reached extends the inactivity timer

Test Completion Indicator: "TCP connections successfully terminated after inactivity" – This test can be marked as complete after the allotted time for expiry has passed and the TCP connection is verified as automatically terminated.

Evaluation Process: This process can be evaluated by inspecting network traffic to note when the TCP connection is created and terminated.

2.30. SYS_PROTEC-6 - User action verification

Steps	Sub-Steps
Create a user from each user group	Create a student user
	Create a faculty user
	Create a registrar user
	Create a FERPA compliance officer user

Attempt each possible task within the system	Attempt all tasks with a user, both allowable and non-allowable Record the state of the system after each system request
Validate that only approved tasks can be performed by a user according to their privileges	Ensure that each action performed is of an allowable type for the user performing the action Record the results of the test

Test Completion Indicator: "User action verification is enforced on all user actions" – This test can be marked as complete when all user actions have been tested with each user type.

Evaluation Process: This process can be evaluated by examining the results of each test round and ensuring that access control is followed in every instance.

Appendix - Requirements Traceability Matrix

Priority	Requirement # by Category	Description	SRS Section	SDS Section	STS Test Case ID
	1	**Logging into the system**	3.3	3.1.5	1
1	1.1	**Validation of user Passwords**	3.3	3.1.5	LOGIN-1
1	1.1.1	**Minimum Password Requirements Met**	3.3	3.1.5	LOGIN-2
1	1.1.2	**Password Hashed and Hash Value Stored for Authentication**	3.3	3.1.5	LOGIN-3

1	1.1.3	**Passwords Must Regularly Be Updated**	3.3	3.1.5	LOGIN-4
2	1.2	**User Sessions will Timeout After a Period of Inactivity**	3.3	2.6	LOGIN-5
2	1.3	**Only Connections from Within the U.S. Should Be Accepted**	3.3	2.1	LOGIN-6
	2	**System Performance**	3.1	3.1.4	2
2	2.1	**Efficient Querying of Database**	3.1	1.1	PERFORM-1
1	2.2	**Sufficient Resources for College User Base**	3.1	1.1	PERFORM-2

2	2.3	**Sufficient Data Throughput for Connections**	3.1	1.1	PERFORM-3
1	2.3.1	**TCP Implemented for User Connections**	3.1	1.1	PERFORM-4
1	2.4	**Critical Sections of Data Modification Protected**	3.1		PERFORM-5
	3	**Data Protection**	3.3	3.1	3
1	3.1	**Encryption of Data at Rest**	3.3	3.1	DATA_PROTEC-1
1	3.2	**Encryption of Data in Transit**	3.3	1.1	DATA_PROTEC-2
1	3.3	**Principle of Least Privilege Applied Throughout Design**	3.3	3.1.2	DATA_PROTEC-3

1	3.4	**Audit Log Maintained of all Data Transformations**	3.3	1.1	DATA_PROTEC-4
	4	**Data Quality**	4	3.1	4
1	4.1	**All Mandatory Fields Contain a Value**	3.2	3.1	DATA_QUAL-1
1	4.2	**Every Student Has a Data Entry**	4	3.1	DATA_QUAL-2
1	4.3	**Data Validation is Performed on All Fields**	4	2	DATA_QUAL-3
1	4.4	**User Records Deleted After Allotted Time**	4	3.1	DATA_QUAL-4
	5	**Legal Compliance**	4	3.1.2	5

1	5.1	**FERPA Regulation Compliance Throughout Design**	4	3.1.2	LEGAL-1
1	5.2	**All U.S. PII Best Practices and Legislation Must Be Followed**	4	3.1.2	LEGAL-2
	6	**Scalability**	4		6
3	6.1	**System Must Be Easily Scalable to Accommodate Student Growth**	4	1.1	SCALABILITY-1
2	6.2	**The system Must Be of Adequate Size to Hold All Student Records Plus Growth Buffer**	4	2.7	SCALABILITY-2
3	6.3	**System Brought Up to Compliance for International Use**	4	3.1.2	SCALABILITY-3

	7	**System Protection**	3.2	3.1	7
1	7.1	**Students Must Be Able to Request Corrections to Their Data**	3.2	3.1.1	SYS_PROTEC-1
1	7.1.1	**Only the Data Owner Can Request Data Changes**	3.2	3.1.1	SYS_PROTEC-2
1	7.2	**Acceptable Use Policy Implemented**	3.2		SYS_PROTEC-3
1	7.3	**DOS Protection**	3.2	1.1	SYS_PROTEC-4
1	7.3.1	**TCP Connections Will Time Out After Inactivity**	3.2	1.1	SYS_PROTEC-5
1	7.4	**User Action Verification**	3.2	3.1.5	SYS_PROTEC-6