Requirements Based Test Cases for University Applicant Determination and Messaging Transactor

University of Minnesota, Twin Cities
CSci5801: Software Engineering
Revision 1.7
03/23/2016

Prepared by Group #17
Krystal Lee 4339496
Qi Wu 5190447
Yuanchen Lu 4620983
Zixiang Ma 4644999

Document Revision History

Rev	Date	Author	Change Description
1.1	3/3/2016	Krystal Lee	Document creation
1.2.0	3/12/2016	Zixiang Ma	Added test cases: E1.1, E1.2, E1.3, E1.4, E4.1, E4.2, E5.1, E5.2, E5.3, E5.4
1.2.1	03/14/2016	Yuanchen Lu	Added test cases: E2.1, E2.2, E2.3, E2.4, E3.1, E3.2
1.3	03/15/2016	Krystal Lee	Organized document, added test modules.
1.4	03/19/2016	Krystal Lee	Added test cases: X1.1, X1.2, X2.1, A6.1, A6.2, A6.3, D1.1, D1.2, D1.3, D1.4, D2.1, D3.1, D3.2, D4.1, D4.2, D5.1, D5.2, D5.3, D6.1, D8.1, D8.2, D9.1, D9.2, D9.3, D9.4, D10.1, C1.1, C2.1, C2.2, C2.3, C2.4, C2.5, C2.6, C5.2, C6.1, C7.1, C7.2 Modified test cases: E6.1.2 E6.1.3
1.5	03/20/2016	Qi Wu	Added test cases:A1.1,A1.2,A3.1, D7.1, D7.2,D11.1,C3.1,C4.1,C5.1 Modified test cases:D1.2,D1.3,D1.4,D2.1,D3.2,D4.1, D4.2,D5.1,D5.2,D5.3,D8.2,D9.3,C2.4,C2.5
1.6	03/23/2016	Zixiang Ma, Yuanchen Lu	Document formatting Modified Traceability Matrix Modified test cases: E1.1, E1.2, E1.3, E1.4, E2.2, E2.3, E2.4, E5.1, E5.2, E5.3, E5.4
1.7	03/23/2016	Krystal Lee	Updated format Added user Traceability Matricies Added test cases: E2.5 Modified test cases: E1.1, E1.2, E1.3, E1.4, E5.1, E5.2, E5.3, E2.4, E3.1, E3.2

This document is originating from Neil Bitzenhofer and DataCard Corporation.

Table of Contents

1. Test Requirements
    1.1 Objective
    1.1 Definitions and Acronyms
    1.3 Traceability Matrix
2. Test Cases
    2.1 Application Tests
    2.2 Validity Check Tests
    2.3 Waitlist Tests
    2.4 Abnormal Situation Tests
    2.5 Filter Tests
    2.6 Screening and Auto-reject Tests
    2.7 Applicant Level User Tests
    2.8 Department Level User Tests
    2.9 College Level User Tests

1. Test Requirements

1.1 Objective

The purpose of the Test Requirements section is to list ALL test requirements, whether explicitly determined from SRS or implicitly determined from experience and product knowledge.. A Test Case Traceability Matrix is provided that simply lists all the test cases by title, and includes a method of tracking what requirements it verifies.

1.1 Definitions and Acronyms

List any technical terms or acronyms used in the document, along with their meanings.

SRS	Software Requirements Specification
TM	Traceability Matrix
U-ADMT	University Application Distribution and Messaging Transactor

1.3 Traceability Matrix

The purpose of a Traceability Matrix is to show which test cases verify which requirements. There are multiple TMs included grouped together by requirement categories. If a requirement has an * next to their name (e.g. *AD3.1) then that means it appears in multiple TMs.

Validity Check Tests

Requirement \ Test Case	Test Case E1.1	Test Case E1.2	Test Case E1.3	Test Case E1.4
ID1				X
ID1.1	X	X	X	

Waitlist Tests

Requirement \ Test Case	Test Case E2.1	Test Case E 2.2	Test Case E2.3	Test Case E2.4	Test Case E2.5
W1	X	X	X	X	
* W1.1	X	X			
W1.2	X				X
* W1.3			X		

W1.4				X	

Abnormal Situation Tests

Requirement \ Test Case	Test Case E3.1	Test Case E3.2
Requirement AB1	x	
Requirement AB2		x

Filter Tests

Requirement \ Test Case	Test Case E4.1	Test Case E4.2
AF1	X	X
AF2	X	X
AC8	X	X

Screen and Auto-Reject Tests

Requirement \ Test Case	Test Case E5.1	Test Case E5.2	Test Case E5.3
S1		X	
S1.1	X	X	X
DU1	X	X	
* AC5	X		

Applicant Level User & Application Creation Tests

Req \ Test Cases	Test Case X1.1	Test Case X1.2	Test Case X2.1	Test Case A1.1	Test Case A1.2	Test Case A1.3	Test Case A2.1	Test Case A2.2	Test Case A2.3	Test Case A2.4	Test Case A3.1
EX1	X	X	X								
EX2						X					
EX3						X					
AD1	X	X									
AD1.1	X										
AD1.2		X									
AD2	X		X								
AD3						X	X	X			
* AD3.1	X	X									
* AD3.2						X	X	X	X		
AD3.4.1								X	X		
AD3.5						X	X				
AP1				X	X						
AP2	X				X						
AP3							X				
AP4						X					
AP5							X	X	X	X	
AP5.1											X
AP5.2							X				
AC2			X								
* W1.3									X		

Administrator Level User & Application Distribution Tests

Req \ Test Cases	Test Case D1.1	Test Case D1.2	Test Case D1.3	Test Case D1.4	Test Case D8.1	Test Case D8.2	Test Case D9.1	Test Case D9.2	Test Case D9.3	Test Case D10.1	Test Case D11.1
AU1	X				X		X			X	
AU2	X				X		X				
AU3											
AU3.1					X	X					
AU3.2										X	
AU3.3					X	X					
AU3.4							X	X	X		
AU3.4.1							X	X			
AU4											X
AD2.1	X	X	X	X							
AD3.1				X							
* AD6										X	
* AD6.1										X	

Faculty Level User & Application Review Tests

Req \ Test Cases	Test Case D2.1	Test Case D3.1	Test Case D3.2	Test Case D4.1	Test Case D4.2	Test Case D5.1	Test Case D5.2	Test Case D5.3	Test Case D6.1	Test Case D7.1	Test Case D7.2
FU1		X		X	X	X	X	X	X		X
FU2	X	X									
FU3		X				X	X	X	X		
FU3.1					X						
FU4											

FU4.1										X	
AD4				X	X	X	X	X			
AD4.1						X					
AD4.2							X				
AD4.3								X			
AD5		X	X								
* AD6									X		

College Level User & Application Status Tests

Req \ Test Cases	Test Case C1.1	Test Case C2.1	Test Case C2.2	Test Case C2.3	Test Case C2.4	Test Case C2.5	Test Case C3.1	Test Case C4.1	Test Case C5.1	Test Case C5.2	Test Case C6.1	Test Case 7.1	Test Case 7.2
AC1													
AC3	X												
AC4		X	X	X		X							
AC4.1							X						
AC4.3													X
* AC5								X					
AC6									X				
AC7										X			
AC8											X		
AC9												X	
* AD3		X	X	X	X	X							
* AD3.2		X											
AD3.3			X										

AD3.4				X	X								
* AD6	X												
* AD6.1	X												
* W1.1			X										

2. Test Cases

2.1 Application Tests

X1.1 Multiple Application Creation for Applicant Level User Using Shared Database

Description Ensure that U-ADMT can correctly access the shared database, and an created application based on application data submitted from an applicant level user.

Test Inputs Shared database initialized with the following:
1. Applicant level user account “rbob”.
2. Application data submitted through the application system by “rbob” containing the following fields:
First Name: Robert
Last Name: Bob
College 1: CSE
Department 1.1: Computer Science and Engineering
Department 1.2: Chemical Engineering
College 2: CLA
Department 2.1: English
Status: Undergraduate

Expected Results 3 (three) applications are created based on the application data with unique application numbers, and are returned. The created applications have an “undecided” status, and applicant level user “rbob” can access the application numbers and statuses. No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “rbob”.

Test Steps

1. Request list of applications for applicant level user “rbob” from U-ADMT.
2. Verify that the list contains 3 (three) applications.
3. Verify that all applications have been assigned a number.
4. Verify that the application numbers are different.
5. Verify that the applications have a status of “undecided”.

Owners: Krystal Lee

X1.2 Valid Application Types

Description Ensure that U-ADMT will correctly create applications with type undergraduate, graduate, or professional only.

Test Inputs

Shared database initialized with the following:
1. Applicant level user account “abob”.
2. Applicant level user account “bbob”.
3. Applicant level user account “cbob”.
4. Applicant level user account “dbob”.
5. College level user account “x”.
6. Application data submitted through the application system by “abob” containing the following field:
<u>Status:</u> Undergraduate

7. Application data submitted through the application system by “bbob” containing the following field:
<u>Status:</u> Graduate

8. Application data submitted through the application system by “cbob” containing the following field:
<u>Status:</u> Professional

9. Application data submitted through the application system by “dbob” containing the following field:
<u>Status:</u> Doctorate

Expected Results

3 (three) applications are created based on input application data, and returned. Each application has a unique number, status of “undecided”, default score of 0 (zero), and has a type of “undergraduate”, “graduate”, or “professional”. An application is not created for the application data with a status of “doctorate”. No exception is thrown.

Dependencies

None.

Initialization

Shared database is loaded, and access to the system has been set through “x”.

Test Steps

1. Request the list of applications from U-ADMT.
2. Verify that the list contains only 3 (three) applications.
3. Verify that all applications have been assigned a number.
4. Verify that the application numbers are different.
5. Verify that the applications have a status of “undecided”.
6. Verify that all applications have a score of 0 (zero).
7. Verify that the only application types are “undergraduate”, “graduate”, or “professional”.
8. Verify that only 1 (one) of the applications has a type of “undergraduate”.
9. Verify that only 1 (one) of the applications has a type of “graduate”.

10. Verify that only 1 (one) of the applications has a type of "professional".

Owners: Krystal Lee

X2.1 Application Distribution Upon Creation for College Level User

Description Ensure that a valid college level user can access a new application created by U-ADMT.

Test Inputs Shared database initialized with the following:
1. College level user account "x".
2. Applicant level user account "rbob".
3. Application data submitted through the application system by "rbob" containing the following fields:
First Name: Robert
Last Name: Bob
College: CSE
Department: Computer Science and Engineering
Status: Undergraduate
GPA: 4.0
ACT: 30
Extracurricular Activities: None
Recommendations: None
Essays: None

Expected Results The application created based on the application data is returned. It has a unique application number, an "undecided" status, and same fields as the application data given in the input. No exception is thrown.

Dependencies Test Case X1.1

Initialization Shared database is loaded, and access to the system has been set through "rbob".

Test Steps

1. Request list of applications from U-ADMT.
2. Verify that the returned list contains 1 (one) application.
3. Verify that the application has been assigned a number.
4. Verify that the application has a status of "undecided".
5. Verify that the application has a type of "undergraduate".
6. Verify that the application has fields correctly matching the fields of the application data in the input.

Owners: Krystal Lee

2.2 Validity Check Tests

E.1.1 Detect Missing Required Information

Description Check input application data from applicant level user and detect invalid data format or value.

Test Inputs Shared database initialized with the following:
First Name:
Last Name: def
College: CSE
Department: ECE
Status: undergraduate
GPA: 3.7
SAT: 1900
ACT:
GRE:
Extracurricular Activities: None
Recommendations: None
Essays: None

Expected Results Input data should not be accepted and system should stop user from proceeding. There is an exception catched.

Dependencies None

Initialization Applicant user account has been activated and has access to the shared database.

Test Steps

1. Check if there is any missing required input.
2. First name is found missing, and there is an exception catched.

Owners Zixiang Ma

E.1.2 Detect Incorrect Data Type.

Description Verify input application data from applicant level user and detect invalid data format or value.

Test Inputs Shared database initialized with the following:
First Name: abc
Last Name: def
College: CSE
Department: ECE
Status: undergraduate
GPA: 3.7
SAT: ord
ACT: 29
GRE:

Extracurricular Activities: None
Recommendations: None
Essays: None

Expected Results The input data should not be accepted and the system should stop user from proceeding. There is an exception catched.

Dependencies None

Initialization Applicant user account has been activated and has access to the shared database.

Test Steps

1. Check if there is any missing required input.
2. Check if there is any input in wrong data format.
3. Invalid data type is found on SAT and error is returned.
4. Do not accept any data and stop user from proceeding. There is an exception catched.

Owners: Zixiang Ma

E.1.3 Detect Out-of-Range Information

Description Check input application data from applicant level user and detect invalid data format or value.

Test Inputs Shared database initialized with the following:
First Name: abc
Last Name: def
College: CSE
Department: ECE
Status: undergraduate
GPA: 3.7
SAT: 21000
ACT: 29
GRE:
Extracurricular Activities: None
Recommendations: None
Essays: None

Expected Results Input data should not be accepted and the system should stop users from proceeding. There is an exception catched.

Dependencies None

Initialization Applicant user account has been activated and logged in successfully.

Test Steps

1. Check if there is any missing required input.
2. Check if there is any input in wrong data format.
3. Check if all input value are in a valid range.
4. Error found: SAT is higher than 2400.

5. Do not accept any input data and stop users from proceeding to next steps.

Owners Zixiang Ma

E.1.4 Accept Valid Input From Applicant Level User

Description Validate all input application data from user and accept them when no invalid input is found.

Test Inputs Shared database initialized with the following:
First Name: abc
Last Name: def
College: CSE
Department: ECE
Status: undergraduate
GPA: 3.7
SAT: 2300
ACT: 29
GRE:
Extracurricular Activities: None
Recommendations: None
Essays: None

Expected Results: System should accept all input data and allow users to proceed to next step.

Dependencies None.

Initialization: Applicant user account has been activated and has access to the shared database.

Test Steps:

1. Check if there is any missing required input.
2. Check if there is any input in wrong data format.
3. Check if all input value are in a valid range.
4. No violation was found in the input and data are accepted.
5. Verify all data is accessible in database

Owners: Zixiang Ma

2.3 Waitlist Tests

E.2.1 Add Application to Waitlist

Description Add an application to waitlist if it has a status of “waitlisted”

Test Inputs Shared database initialized with the following:
An application with status of “waitlisted”

Expected Results A queue of waitlisted applications is returned. No exception is thrown.

Dependencies None.

Initialization An application has been reviewed and assigned with temporary admission status.

Test Steps

1. Identify which waitlist queue (undergraduate, graduate, or professional) the application belongs to.
2. Add the application to the waitlist queue based on its score
3. Verify that the application has been added to the waitlist queue

Owners: Yuanchen Lu

E.2.2 Sort Applications in Waitlist

Description Sort applications in a waitlist in decreasing order

Test Inputs Shared database initialized with the following:

1. All applications of type "undergraduate" with status "waitlisted"
2. The "undergraduate" waitlist.

Expected Results A sorted waitlist that matches the "undergraduate" waitlist.

Dependencies None

Initialization There is a waitlist created

Test Steps

1. List out the scores assigned to each application
2. Sort applications in decreasing order
3. Verify that the waitlist sorted matches the "undergraduate" waitlist in the database.

Owners Yuanchen Lu and Krystal Lee

E.2.3 Remove Application From Waitlist

Description Remove the first entry in the waitlist queue

Test Inputs Waitlist queue

Expected Results The first entry in the queue returned. If no entry exists in the queue, throw an exception.

Dependencies None

Initialization All application results have been sent out. There are admission openings

Test Steps

4. Check if the queue is empty. Throw an exception if yes.
5. Pop the first entry in the queue
6. Make the second entry the head of the queue
7. Verify that no other application scores in the queue are greater than the retrieved application's

Owners: Yuanchen Lu

E2.4 Waitlist Removal Disabled

Description Ensure that if waitlist removal is disabled, U-ADMT does not remove applications from the waitlist.

Test Inputs Shared database initialized with the following:
1. College level user account “x”.
2. Application #1234 that contains the following fields:
<u>Status:</u> Waitlisted
<u>Type:</u> Undergraduate
<u>Department:</u> Computer Science and Engineering
<u>Score:</u> 85

3. Waitlist for “undergraduate” applications to “Computer Science and Engineering” with the following fields:
<u>Removal:</u> Disabled
<u>Applications in decreasing order:</u> #1234

Expected Results The intended test operation does not go through, and there is an exception catched.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “x”.

Test Steps

1. Request application #1234.
2. Verify that for each field of the application the information is correct.
3. Change the application status from “waitlisted” to “accepted”.
4. Verify that U-ADMT does not allow the status to be changed to “accepted”, and there is an exception catched.

Owners: Krystal Lee

E2.5 Multi-Score Waitlist Add.

Description Ensure that if an application is added to a waitlist where the score is the same another score in the waitlist then U-ADMT adds it below the application with the same score.

Test Inputs Shared database initialized with the following:
1. College level user account “x”.
2. Application #1234 that contains the following fields:
<u>Status:</u> Waitlisted
<u>Type:</u> Undergraduate
<u>Department:</u> Computer Science and Engineering
<u>Score:</u> 85

2. Application #2345 that contains the following fields:
Status: Undecided
Type: Undergraduate
Department: Computer Science and Engineering
Score: 85

3. Waitlist for "undergraduate" applications to "Computer Science and Engineering" with the following fields:
Removal: Disabled
Applications in decreasing order: #1234

Expected Results The application #2345 status is updated to "waitlisted", and it is added to the waitlist below application #1234.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through "x".

Test Steps

1. Request waitlist for "undergraduate" to "Computer Science and Engineering" from U-ADMT.
2. Verify that the returned waitlist contains 1 (one) application.
3. Verify that the application returned is #1234.
4. Change the status of application #2345 to "waitlisted".
5. Request waitlist for "undergraduate" to "Computer Science and Engineering" from U-ADMT.
6. Verify that the returned waitlist contains 2 (two) applications.
7. Verify that the applications returned are #1234 and #2345, in that order.

Owners: Krystal Lee

2.4 Abnormal Situation Tests

E.3.1 Memory Overflow Handler

Description Ensure that U-ADMT can correctly handle memory overflow by limiting limited access.

Test Inputs Shared database.

Expected Results System handles memory overflow and prevents further system malfunction.

Dependencies None

Initialization: Shared database is loaded, and is accessible.

Test Steps:

1. Overflow the memory.
2. Verify that the system acknowledges such memory overflow.
3. Verify that the system limits access to applications.

4. Clear the memory.
5. Verify system fixes the memory overflow issue.

Owners: Yuanchen Lu and Krystal Lee

E.3.2 Network Disconnect Handler

Description Ensure that U-ADMT can correctly handle a network disconnection by attempting to reconnect regularly.

Test Inputs Shared database.

Expected Results System handles network disconnect issue and prevents further system malfunction.

Dependencies None

Initialization Shared database is loaded, and is accessible.

Test Steps

1. Disconnect the network.
2. Verify that the system acknowledges a network disconnect.
3. Verify that the system attempts to reconnect to server every 10 seconds.
4. Reconnect to network.
5. Verify system fixes the network disconnect issue,

Owners Yuanchen Lu

2.5 Filter Tests

E. 4.1 Filter with valid criteria

Description Filter function only returns those application that satisfy a criteria.

Test Inputs Application information from database
Filter criteria type: GPA
Operation: >
Filter criteria value: 3.5

Expected Results Only return applications containing GPA higher than 3.5

Dependencies None

Initialization All application information has been stored in database and available for access.

Test Steps

1. Verify the criteria type is available in database as a data attribute.
2. Visit each application entry and compare its attributes with filter criteria.
3. If the entry match filter criteria, the system should keep it to return to the user.

4. If the entry fail to match the filter criteria, the system should remove it from list of applications to return.
5. Verify the number of filtered result is higher than 0 and lower than the total number of application.
6. All information should still be available in database after filtering.

Owners Zixiang Ma

E. 4.2 Filter With Invalid Criteria

Description Filter function only displays those application that satisfies a criteria.

Test Inputs Application information from database
Criteria type: salary
Operation: >
Criteria value: 3.5

Expected Results The intended test operation does not go through, and there is an exception catched.

Dependencies None

Initialization All application information has been stored in database and available for access.

Test Steps

1. Verify filter criteria type is an available attributes in database.
2. Verify that no matched attribute is found, and there is an exception catched..
3. Verify no change has been made to the database.

Owners Zixiang Ma

2.6 Screening and Auto-reject Tests

E. 5.1 Screen Application With Minimum Threshold.

Description System should select application with attributes lower than associated threshold value.

Test Inputs Shared database initialized with the following:
GPA threshold: 3.0
SAT threshold: 1600
ACT threshold: 22
GRE threshold: 310
All application information related with certain department.

Expected Results Application with any attributes lower than associated threshold are selected for further process.

Dependencies E1.1, E1.2, E1.3, E1.4

Initialization All application information has been stored in database and available for access.

Test Steps

1. Verify input minimum threshold is valid
2. Verify the number of application selected is greater than zero but lower than the number of total application.
3. Verify one or several of the attributes in the application selected is lower minimum threshold.
4. Verify no change is made to the database.

Owners Zixiang Ma

E. 5.2 Auto-Reject Application Based on Screening Results.

Description Application selected in the screening process will be given the option to be rejected automatically.

Test Inputs Shared database initialized with the following:
Application screened by minimum thresholds

Expected Results Application selected in the screening has its recommendation changed to rejected and default score to 0.

Dependencies E5.1

Initialization All application information has been stored in database and available for access.

Test Steps

1. Follow E5.1
2. Choose to auto-reject the applications.
3. Verify all auto-rejected application has recommendation of "rejected".
4. Verify the number of unprocessed application is reduced by the number of application auto-rejected.

Owners Zixiang Ma and Krystal Lee

E. 5.3 Auto-Reject Application Deletes Rejected Application From the Database.

Description Application information should be deleted when it is rejected.

Test Inputs None

Expected Results Rejected applications are deleted from the database.

Dependencies Auto-reject

Initialization All application information has been stored in database and available for access.

Test Steps

1. Follow E5.2
2. Verify rejected application has been deleted from the database.

3. Verify the deleted applications are inaccessible.
4. Verify the existing applications have attributes higher than the minimum screening threshold.

Owners Zixiang Ma

2.7 Applicant Level User Tests

A1.1 Applicant level user - Account Access

Description Ensure that applicant level user can access to function in its user level after access to the system.

Test Inputs 10 Applicant level user accounts initialized in the database.

Expected Results All the function for applicant level user is available for each account.

Dependencies None.

Initialization Applicant account has already registered and stored in the database.

Test Steps

1. Enter the correct applicant level username and password for the test applicant account.
2. Check if all the function, including application status access, decision notification, and decision response is available.

Owners: Qi Wu

A1.2 Application level user - Application Status Access

Description: Ensure that applicant level user can access the application number and application status correctly and confirm the status.

Test Inputs Several applicant accounts, which have different Application Status in the shared database, including accepted, rejected, waitlisted and undecided are initialized in the database.

Expected Results Correct application status is shown.

Dependencies None.

Initialization Applicant account has already registered and stored in the database, there is a status for each the application.

Test Steps

1. Enter the correct applicant level username and password for each test applicant account.
2. Check if the status given is the same as the status in the database.

Owners Qi Wu

A1.3 Applicant Level User Decision Notification via Messaging Transactor

Description Ensure that U-ADMT triggers email to be sent from email system to applicant level user when their corresponding application status is changed from undecided.

Test Inputs Shared database initialized with the following:
1. College level user account "x".
2. Applicant level user account "rbob" that contains the following field:
Username: rbob
Email: rbob@gmail.com

2. Application #1234 that is associated with applicant level user "rbob", and contains the following fields:
Status: Accepted

3. Acceptance Threshold for "undergraduate" and "Computer Science and Engineering": 10

Expected Results U-ADMT triggers an email notification to be sent from email system to "rbob@gmail.com".

Dependencies Test Case C2.1

Initialization Shared database is loaded, and access to the system has been set through "x".

Test Steps

1. Follow Test Case C2.1
2. Request application #1234 from U-ADMT.
3. Verify that the information in the returned application is correct.
4. Verify that U-ADMT triggered email to be sent from email system to email address associated with applicant level user "rbob".

Owners Qi Wu and Krystal Lee

A2.1 Applicant Level User Change Multiple Application Status to Confirmed

Description Ensure that a valid applicant level user can change the application status of an application from "accepted" to "confirmed", and that if another application has a status of "confirmed" associated with the account the status will be changed from "confirmed" to "accepted".

Test Inputs Shared database initialized with the following:
1. Applicant level user account "rbob".
2. Application #1234 that is associated with applicant level user "rbob", and contains the following field:
Status: Accepted

3. Application #5678 that is associated with applicant level user “rbob”, and contains the following field:
Status: Confirmed

Expected Results The status of application #1234 is changed to “confirmed” from “accepted”, and the status of application #5678 is changed to “accepted” from “confirmed”. No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “rbob”.

Test Steps

1. Request the list of applications for applicant level user “rbob” from U-ADMT.
2. Verify that the list contains 2 (two) applications.
3. Verify that the returned applications are #1234 and #5678.
4. Verify that for each field of each returned application the information is correct.
5. Change the application status of application #1234 from “accepted” to “confirmed”.
6. Verify that the status of application #1234 is “confirmed”.
7. Verify that the status of application #5678 is “accepted”.

Owners Qi Wu and Krystal Lee

A2.2 Applicant Level User Change Application Status to Rejected

Description Ensure that a valid applicant level user can change the application status of an application from “accepted” to “rejected”.

Test Inputs Shared database initialized with the following:
1. College level user account “x”.
2. Applicant level user account “rbob”.
3. Application #1234 that is associated with applicant level user “rbob”, and contains the following field:
Status: Accepted
Type: Undergraduate
Department: Computer Science and Engineering

4. Application #2345 that contains the following fields:
Status: Waitlisted
Type: Undergraduate
Department: Computer Science and Engineering

5. Application #3456 that contains the following fields:
Status: Waitlisted

Type: Undergraduate
Department: Computer Science and Engineering

6. Waitlist for “undergraduate” applications to “Computer Science and Engineering” with the following fields:
Removal: Enabled
Applications in decreasing order:

1. Application #2345
2. Application #3456

Expected Results

The status of application #1234 is changed to “rejected” from “accepted”. No exception is thrown.

Dependencies

None.

Initialization

Shared database is loaded, and access to the system has been set through “rbob”.

Test Steps

1. Request the list of applications for applicant level user “rbob” from U-ADMT.
2. Verify that the list contains 1 (one) application.
3. Verify that the returned applications is #1234.
4. Verify that the status of the application is “accepted”
5. Change the application status from “accepted” to “rejected”.
8. Verify that the status of the application is “rejected”.

Owners

Krystal Lee

A2.3 Applicant Level User Change Application Status to Rejected With Waitlist Removal

Description

Ensure that if a valid applicant level user changes an application status from “accepted” to “rejected”, and if waitlist removal is enabled U-ADMT will remove an application from the waitlist matching the rejected application’s department and type.

Test Inputs

Shared database initialized with the following:
1. College level user account “x”.
2. Applicant level user account “rbob”.
3. Application #1234 that is associated with applicant level user “rbob”, and contains the following field:
Status: Rejected
Type: Undergraduate
Department: Computer Science and Engineering

4. Application #2345 that contains the following fields:
Status: Accepted

Type: Undergraduate
Department: Computer Science and Engineering

5. Application #3456 that contains the following fields:
Status: Waitlisted
Type: Undergraduate
Department: Computer Science and Engineering

6. Waitlist for “undergraduate” applications to “Computer Science and Engineering” with the following fields:
Removal: Enabled
Applications in decreasing order:

1. Application #3456

Expected Results The applications are returned with the expected values given in the input. The waitlist is returned with the expected application given in the input. # No exception is thrown.

Dependencies Test Case 1.1

Initialization: Shared database is loaded, and access to the system has been set through “x”.

Test Steps:

1. Follow Test Case 1.1
2. Request Application #1234 from U-ADMT.
3. Verify that for each field of application #1234 the information is correct.
4. Request Application #2345 from U-ADMT.
5. Verify that for each field of application #2345 the information is correct.
6. Request waitlist for “undergraduate” to “Computer Science and Engineering Department” from U-ADMT.
7. Verify that the returned waitlist contains 1 (one) application.
8. Verify that the application returned is #3456.
9. Verify that for each field of application #3456 the information is correct.

Owners: Krystal Lee

A2.4 Invalid Applicant Level User Change Application Status

Description Ensure that U-ADMT does not allow a valid applicant level user to change the application status of an application from “accepted” to something other than “rejected” or “confirmed”.

Test Inputs Shared database initialized with the following:
1. Applicant level user account “rbob”.

2. Application #1234 that is associated with applicant level user “rbob”, and contains the following field:
Status: Accepted

Expected Results The intended test operation does not go through, and there is an exception catched

Dependencies None.

Initialization: Shared database is loaded, and access to the system has been set through “rbob”.

Test Steps:

1. Request the list of applications for applicant level user “rbob” from U-ADMT.
2. Verify that the list contains 1 (one) applications.
3. Verify that the returned application is #1234.
4. Verify that for each field of the application the information is correct.
5. Change the application status from “accepted” to “ok”.
9. Verify that U-ADMT does not allow the status to be changed to “ok”, and there is an exception catched.

Owners: Krystal Lee and Qi Wu

A3.1 Applicant Level User - Respond to Decision Failure

Description Ensure that application status will automatically change to rejected when applicant level user fails to respond before the deadline.

Test Inputs Several applicant account in the shared database, with the application decision status accepted, deadline is set about 5 minutes later.

Expected Results The application is changed to rejected automatically.

Dependencies None.

Initialization: Applicant account has already registered and stored in the database, the application status is set properly, the deadline is set

Test Steps:

1. Wait to the deadline pass.
1. Check if all applications with accepted decision have automatically changed to rejected.

Owners: Qi Wu

2.8 Department Level User Tests

D1.1 Administrator Level User Assigning Applications for Review

Description Ensure that a valid administrator level user can assign an application with undecided status for review to 3 (three) faculty level users from the same department.

Test Inputs Shared database initialized with the following:
1. Administrator level user account “ad” that contains the following field:
Department: Computer Science and Engineering

2. Faculty level user account “af” that contains the following field:
Department: Computer Science and Engineering

3. Faculty level user account “bf” that contains the following field:
Department: Computer Science and Engineering

4. Faculty level user account “cf” that contains the following field:
Department: Computer Science and Engineering

5. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Score: 0
Assigned To: None
Review Deadline: None

Expected Results Application #1234 is assigned to “af”, “bf”, and “cf” with a review deadline of 2 (two) weeks. No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “ad”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Assign “af” to review the application.
4. Verify that “af” has been assigned to review the application.
5. Assign “bf” to review the application.
6. Verify that “bf” has been assigned to review the application.
7. Assign “cf” to review the application.
8. Verify that “cf” has been assigned to review the application.
9. Assign 2 (two) weeks as review deadline for the application.
10. Verify that 2 (two) weeks has been assigned as review deadline for the application.

Owners: Krystal Lee

D1.2 Invalid Administrator Level User Signing Applications for Review - Oversigning

Description Ensure that a valid administrator level user cannot assign more than 3 (three) faculty level users to an application for review.

Test Inputs Shared database initialized with the following:
1. Administrator level user account “ad” that contains the following field:
Department: Computer Science and Engineering

2. Faculty level user account “af” that contains the following field:
Department: Computer Science and Engineering

3. Faculty level user account “bf” that contains the following field:
Department: Computer Science and Engineering

4. Faculty level user account “cf” that contains the following field:
Department: Computer Science and Engineering

5. Faculty level user account “df” that contains the following field:
Department: Computer Science and Engineering

6. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Department: Computer Science and Engineering
Score: 0
Assigned To: “af”, “bf”, “cf”
Review Deadline: 2 weeks

Expected Results The intended test operation does not go through, and there is an exception catched

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “ad”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Assign “df” to review the application.
3. Verify that U-ADMT does not allow “df” to be assigned to review the application, and there is an exception catched.

Owners: Krystal Lee and Qi Wu

D1.3 Invalid Administrator Level User Signing Applications for Review - Wrong Department

Description Ensure that a valid administrator level user cannot assign a faculty level user to an application for review if the faculty level user is not in the same department.

Test Inputs Shared database initialized with the following:
1. Administrator level user account “ad” that contains the following field:
Department: Computer Science and Engineering

2. Faculty level user account “af” that contains the following field:
Department: Chemical Engineering

3. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Department: Computer Science and Engineering
Score: 0
Assigned To: None
Review Deadline: None

Expected Results The intended test operation does not go through, and there is an exception catched.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “ad”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Assign “af” to review the application.
3. Verify that U-ADMT does not allow “af” to be assigned to review the application, and there is an exception catched.

Owners: Krystal Lee and Qi Wu

D1.4 Invalid Administrator Level User Signing Applications for Review - Not Undecided

Description Ensure that a valid administrator level user cannot assign a faculty level user to an application for review if the application does not have a status of undecided.

Test Inputs Shared database initialized with the following:
1. Administrator level user account “ad” that contains the following field:
Department: Computer Science and Engineering

2. Faculty level user account “af” that contains the following field:
Department: Computer Science and Engineering

3. Application #1234 that contains the following fields:
Status: Accepted
Type: Undergraduate
Department: Computer Science and Engineering

Expected Results The intended test operation does not go through, and there is an exception catched

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through "ad".

Test Steps

1. Request application #1234 from U-ADMT.
2. Assign "af" to review the application.
3. Verify that U-ADMT does not allow "af" to be assigned to review the application, and there is an exception catched.

Owners: Krystal Lee and Qi Wu

D2.1 Invalid Faculty Level User Review Unassigned Application

Description Ensure that a valid faculty level user cannot review an application they are not assigned to.

Test Inputs Shared database initialized with the following:
1. Faculty level user account "af".
2. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Assigned To: None

Expected Results The intended test operation does not go through, and there is an exception catched

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through "af".

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Give application #1234 a recommendation of "accept".
4. Verify that U-ADMT does not allow the adding of a recommendation, and there is an exception catched.

Owners: Krystal Lee and Qi Wu

D3.1 Faculty Level User Add Recommendation to Assigned Application

Description Ensure that a valid faculty level user can review an application they are assigned to by adding a recommendation of “accept”, “reject”, or “waitlist”.

Test Inputs Shared database initialized with the following:
1. Faculty level user account “af”.
2. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Assigned To: “af”
Recommendation From “af”: None

3. Application #5678 that contains the following fields:
Status: Undecided
Type: Graduate
Assigned To: “af”
Recommendation From “af”: None

4. 3. Application #9012 that contains the following fields:
Status: Undecided
Type: Professional
Assigned To: “af”
Recommendation From “af”: None

Expected Results Application #1234 is given a recommendation of “accept” from “af”. Application #5678 is given a recommendation of “reject” from “af”. Application #9012 is given a recommendation of “waitlist from “af””. No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “af”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Give application #1234 a recommendation of “accept”.
4. Verify that the recommendation from “af” for #1234 is “accept”.
5. Request application #5678 from U-ADMT.
6. Verify that the information in the returned application is correct.
7. Give application #5678 a recommendation of “reject”.
8. Verify that the recommendation from “af” for #5678 is “reject”.
9. Request application #9012 from U-ADMT.
10. Verify that the information in the returned application is correct.
11. Give application #9012 a recommendation of “waitlist”.
12. Verify that the recommendation from “af” for #9012 is “waitlist”.

Owners: Krystal Lee

D3.2 Invalid Faculty Level User Add Recommendation to Assigned Application

Description Ensure that a valid faculty level user cannot add a recommendation that is not “accept”, “reject”, or “waitlist” to an application they are assigned to review.

Test Inputs Shared database initialized with the following:
1. Faculty level user account “af”.
2. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Assigned To: “af”
Recommendation From “af”: None

Expected Results The intended test operation does not go through, and there is an exception catched

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “af”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Give application #1234 a recommendation of “ok”.
4. Verify that U-ADMT does not allow “ok” to be given as a recommendation, and there is an exception catched.

Owners: Krystal Lee and Qi Wu

D4.1 Invalid Faculty Level User Add Score to Assigned Application - Past Max

Description Ensure that a valid faculty level user cannot give an assigned application a score larger than the maximum of 100 (hundred).

Test Inputs Shared database initialized with the following:
1. Faculty level user account “af”.
2. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Assigned To: “af”
Score From “af”: 0
GPA Score From “af”: 0
Extracurricular Score From “af”: 0

Recommendation Score From “af”: 0
ACT/SAT Score From “af”: 0
Essay Score From “af”: 0

Expected Results The intended test operation does not go through, and there is an exception catched

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “af”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Give the application a GPA score of 25 (twenty-five).
4. Verify that the score for the application is 25 (twenty-five).
5. Give the application a Extracurricular score of 20 (twenty).
6. Verify that the score for the application is 45 (forty-five).
7. Give the application a Recommendation score of 15 (fifteen).
8. Verify that the score for the application is 60 (sixty).
9. Give the application a ACT/SAT score of 25 (twenty-five).
10. Verify that the score for the application is 85 (eighty-five).
11. Give the application an Essay score of 16 (sixteen).
12. Verify that U-ADMT does not allow the score to be larger than 100 (hundred), and there is an exception catched.

Owners: Krystal Lee and Qi Wu

D4.2 Invalid Faculty Level User Add Score to Assigned Application - Past Min

Description Ensure that a valid faculty level user cannot give an assigned application a score smaller than the minimum of 1 (one).

Test Inputs Shared database initialized with the following:
1. Faculty level user account “af”.
2. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Assigned To: “af”
Score From “af”: 0
GPA Score From “af”: 0
Extracurricular Score From “af”: 0
Recommendation Score From “af”: 0
ACT/SAT Score From “af”: 0
Essay Score From “af”: 0

Expected Results The intended test operation does not go through, and there is an exception catched

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through "af".

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Give the application a GPA score of 0 (zero).
4. Verify that U-ADMT does not allow the score to be smaller than 1 (one), and there is an exception catched.

Owners: Krystal Lee and Qi Wu

D5.1 Faculty Level User Add Score to Assigned Undergraduate Application

Description Ensure that a valid faculty level user can review an undergraduate application they are assigned to by giving it a score based on GPA, extracurriculars, recommendations, ACT/SAT, and essays.

Test Inputs Shared database initialized with the following:
1. Faculty level user account "af".
2. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Assigned To: "af"
Score From "af": 0
GPA Score From "af": 0
Extracurricular Score From "af": 0
Recommendation Score From "af": 0
ACT/SAT Score From "af": 0
Essay Score From "af": 0

Expected Results Application #1234 is given a score from "af". No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through "af".

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Give the application a GPA score of 25 (twenty-five).
4. Verify that the GPA score for the application from "af" is 25 (twenty-five).
5. Give the application a Extracurricular score of 20 (twenty).

6. Verify that the Extracurricular score for the application from “af” is 45 (forty-five).
7. Give the application a Recommendation score of 15 (fifteen).
8. Verify that the Recommendation score for the application from “af” is 60 (sixty).
9. Give the application a ACT/SAT score of 25 (twenty-five).
10. Verify that the ACT/SAT score for the application from “af” is 85 (eighty-five).
11. Give the application an Essay score of 15 (fifteen).
12. Verify that the Essay score for application from “af” is 15.
13. Verify that the total score for the application from “af” is calculated correctly from the formula given in the requirement.

Owners: Krystal Lee and Qi Wu

D5.2 Faculty Level User Add Score to Assigned Graduate Application

Description Ensure that a valid faculty level user can review a graduate application they are assigned to by giving it a score based on GPA, research experience, recommendations, GRE, and essays.

Test Inputs Shared database initialized with the following:
1. Faculty level user account “af”.
2. Application #1234 that contains the following fields:
Status: Undecided
Type: Graduate
Assigned To: “af”
Score From “af”: 0
GPA Score From “af”: 0
Research Score From “af”: 0
Recommendation Score From “af”: 0
GRE Score From “af”: 0
Essay Score From “af”: 0

Expected Results Application #1234 is given a score from “af”. No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “af”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Give the application a GPA score of 30 (thirty).
4. Verify that the GPA score for the application from “af” is 30 (thirty).

5. Give the application a Research score of 20 (twenty).
6. Verify that the Research score for the application from “af” is 50 (fifty).
7. Give the application a Recommendation score of 70 (seventy).
8. Verify that the Recommendation score for the application from “af” is 70 (seventy).
9. Give the application a GRE score of 85 (eighty-five).
10. Verify that the GRE score for the application from “af” is 85 (eighty-five).
11. Give the application an Essay score of 15 (fifteen).
12. Verify that the Essay score for the application from “af” is 15(fifteen).
13. Verify that the score is calculated correctly from the formula given by the requirement.

Owners: Krystal Lee and Qi Wu

D5.3 Faculty Level User Add Score to Assigned Professional Application

Description Ensure that a valid faculty level user can review a professional application they are assigned to by giving it a score based on GPA, work experience, recommendations, and essays.

Test Inputs Shared database initialized with the following:
1. Faculty level user account “af”.
2. Application #1234 that contains the following fields:
Status: Undecided
Type: Professional
Assigned To: “af”
Score From “af”: 0
GPA Score From “af”: 0
Work Score From “af”: 0
Recommendation Score From “af”: 0
Essay Score From “af”: 0

Expected Results Application #1234 is given a score of 100 from “af”. No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “af”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Give the application a GPA score of 25 (twenty-five).
4. Verify that the GPA score for the application from "af" is 25 (twenty-five).
5. Give the application a Work score of 50 (fifty).
6. Verify that the Work score for the application from "af" is 50 (fifty).
7. Give the application a Recommendation score of 75 (seventy-five).
8. Verify that the score for the application from "af" is 75 (seventy-five).
9. Give the application an Essay score of 25 (twenty-five).
10. Verify that the Essay score for the application from "af" is 25(twenty-five).
11. Verify that the total score for the application from "af" is calculated correctly from the formula given by the requirement

Owners: Krystal Lee and Qi Wu

D6.1 Faculty Level User Add Comment to Assigned Application

Description Ensure that a valid faculty level user can review an application they are assigned to by adding an ASCII text comment.

Test Inputs Shared database initialized with the following:
1. Faculty level user account "af".
2. Application #1234 that contains the following fields:
<u>Status:</u> Undecided
<u>Type:</u> Undergraduate
<u>Assigned To:</u> "af"

Expected Results Application #1234 is given a comment of "good application" from "af". No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through "af".

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Give the application a comment of "good application".
4. Verify that the application has a comment from "af" that is "good application".

Owners: Krystal Lee

D7.1 Faculty Level User Failing to Submit Application

Description Ensure that when the faculty level user fails to submit the application before the deadline, the system will change the application score to 0 and recommendation to rejected.

Test Inputs Shared database initialized with the following:
1. Faculty level user account “af”.
2. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Assigned To: “af”
Score From “af”: 0
Recommendation From “af”: None
Deadline: 5 min
This application has not submitted by the faculty level user.

Expected Results In application #1234 “af” is removed from the “assigned to” field. The score is set to a default value of 0, and a recommendation of “reject” is assigned.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “af”.

Test Steps

1. Wait until the deadline passed.
2. Verify that score field is set to default value of 0.
3. Verify that Recommendation is set to “reject”.
4. Verify if “af” is removed from Assign_To, and the application is submitted to administrator level user.

Owners: Qi Wu and Krystal Lee

D7.2 Faculty Level User Submit Application

Description Ensure that when the faculty level user submits the application the application will return to the administrator level user.

Test Inputs Shared database initialized with the following:
1. Faculty level user account “af”.
2. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Assigned To: “af”
This application has not submitted by the faculty level user.

Expected Results Application #1234 can be accessed by the administrator level user with faculty “af” recommendation and score. No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “af”.

Test Steps

1. Submit the application.
2. Verify the application is returned to administrator level user and all the score fields, deadline, Assign_To field are correct.

Owners: Qi Wu

D8.1 Administrator Level User Collate Scores for Assigned Application

Description Ensure that a valid administrator level user can collate scores for an assigned application by averaging the scores from the 3 (three) faculty level users, and where a default score of 0 (zero) is not counted.

Test Inputs Shared database initialized with the following:
1. Administrator level user account “ad”.
2. Faculty level user account “af”.
3. Faculty level user account “bf”.
4. Faculty level user account “cf”.
5. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Score: 0
Assigned To: “af”, “bf”, cf”
Score From “af”: 100
Score From “bf”: 0
Score From “cf”: 50

Expected Results Application #1234 is given a collated score of 75 (seventy-five). No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “ad”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Collate the application scores from the 3 (three) assigned faculty level users.
4. Verify that the application score is 75 (seventy-five).

Owners: Krystal Lee

D8.2 Invalid Administrator Level User Collate Scores for Assigned Application

Description Ensure that a valid administrator level user cannot collate scores for an application if it has not been assigned to 3 (three) faculty level users.

Test Inputs Shared database initialized with the following:
1. Administrator level user account “ad”.
2. Faculty level user account “af”.
3. Faculty level user account “bf”.
4. Application #1234 that contains the following fields:
<u>Status:</u> Undecided
<u>Type:</u> Undergraduate
<u>Score:</u> 0
<u>Assigned To:</u> “af”, “bf”
<u>Score From “af”:</u> 100
<u>Score From “bf”:</u> 0

Expected Results The intended test operation does not go through, and there is an exception catched.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “ad”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Collate the application scores from the 2 (two) assigned faculty level users.
5. Verify that U-ADMT does not collate the scores from “af” and “bf”, and there is an exception catched.

Owners: Krystal Lee and Qi Wu

D9.1 Administrator Level User Majority Recommendation for Assigned Application

Description Ensure that a valid administrator level user can take majority vote for an application from the recommendations of the 3 (three) assigned faculty level users where the recommendation that is made at least 2 (two) times is chosen.

Test Inputs Shared database initialized with the following:
1. Administrator level user account “ad”.
2. Faculty level user account “af”.
3. Faculty level user account “bf”.
4. Faculty level user account “cf”.
5. Application #1234 that contains the following fields:
<u>Status:</u> Undecided

Type: Undergraduate
Recommendation: 0
Assigned To: “af”, “bf”, cf”
Recommendation From “af”: Accept
Recommendation From “bf”: Accept
Recommendation From “cf”: Reject

Expected Results Application #1234 is given a recommendation of “accept”. No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “ad”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Take majority vote on the application recommendations from the 3 (three) assigned faculty level users.
4. Verify that the application recommendation is “accept”.

Owners: Krystal Lee

D9.2 Administrator Level User Majority Recommendation for Assigned Application - Tie

Description Ensure that a valid administrator level user can take majority vote for an application from the recommendations of the 3 (three) assigned faculty level users where if there is a tie the recommendation will be “waitlist”.

Test Inputs Shared database initialized with the following:
1. Administrator level user account “ad”.
2. Faculty level user account “af”.
3. Faculty level user account “bf”.
4. Faculty level user account “cf”.
5. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Recommendation: 0
Assigned To: “af”, “bf”, cf”
Recommendation From “af”: Accept
Recommendation From “bf”: Waitlist
Recommendation From “cf”: Reject

Expected Results Application #1234 is given a recommendation of “waitlist”. No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through "ad".

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Take majority vote on the application recommendations from the 3 (three) assigned faculty level users.
4. Verify that the application recommendation is "waitlist".

Owners: Krystal Lee

D9.3 Invalid Administrator Level User Majority Recommendation for Assigned Application

Description Ensure that a valid administrator level user cannot take majority recommendation for an application if it has not been assigned to 3 (three) faculty level users.

Test Inputs Shared database initialized with the following:
1. Administrator level user account "ad".
2. Faculty level user account "af".
3. Faculty level user account "bf".
4. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Recommendation: 0
Assigned To: "af", "bf"
Recommendation From "af": Accept
Recommendation From "bf": Accept

Expected Results The intended test operation does not go through, and there is an exception catched

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through "ad".

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Take majority vote on the application recommendations from the 3 (three) assigned faculty level users.
4. Verify that U-ADMT does not take majority recommendation from "af" and "bf", and there is an exception catched.

Owners: Krystal Lee and Qi Wu

D10.1 Administrator Level User Reply Comment

Description Ensure that a valid administrator level user can reply to comments made by faculty level users with their own ASCII text comment.

Test Inputs Shared database initialized with the following:
1. Administrator level user account “ad”.
2. Faculty level user account “af”.
3. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Assigned To: “af”
Comment From “af”: “good application”

Expected Results Application #1234 is given a reply comment of “bad application actually” from “ad” to the comment from “af”. No exception is thrown.

Dependencies Test Case 6.1

Initialization Shared database is loaded, and access to the system has been set through “ad”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Reply to the comment from “af” with the comment “bad application actually”.
4. Verify that the application has a reply comment from “ad” to “af” that is “bad application actually”.

Owners: Krystal Lee

D11.1 Administrator Level User submit application

Description Ensure that a valid administrator level user can submit the application and college level user can view the submitted application.

Test Inputs Shared database initialized with the following:
1. Administrator level user account “ad”.
2. Faculty level user account “af”.
3. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Assigned To: “af”
Comment From “af”: “good application”

Expected Results Application #1234 can be viewed by the college level user after submission.No exception is thrown.

Dependencies

Initialization Shared database is loaded, and access to the system has been set through “ad”.

Test Steps

1. “Ad” submit the application.
5. Verify that the application can be viewed by college level user.

Owners: Qi Wu

2.9 College Level User Tests

C1.1 College Level User Reply Comment

Description Ensure that a valid college level user can reply to comments made by department level users with their own ASCII text comment.

Test Inputs Shared database initialized with the following:
1. College level user account “x”.
2. Administrator level user account “ad”.
3. Faculty level user account “af”.
4. Application #1234 that contains the following fields:
<u>Status:</u> Undecided
<u>Type:</u> Undergraduate
<u>Comment From “af”:</u> “good application”
<u>Reply From “ad” to “af”:</u> “bad application actually”

Expected Results Application #1234 is given a reply comment of “no it is good” from “x” to the reply comment from “ad”. No exception is thrown.

Dependencies Test Case 10.1

Initialization Shared database is loaded, and access to the system has been set through “x”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Reply to the reply comment from “ad” with the comment “no it is good”.
6. Verify that the application has a reply comment from “x” to “ad” that is “no it is good”.

Owners: Krystal Lee

C2.1 College Level User Change Final Application Status to Accepted

Description Ensure that a valid college level user can change the application status of an application from “undecided” to “accepted”.

Test Inputs Shared database initialized with the following:
1. College level user account “x”.
2. Applicant level user account “rbob” that contains the following field:
<u>Username:</u> rbob
<u>Email:</u> rbob@gmail.com

3. Application #1234 that is associated with applicant level user “rbob”, and contains the following fields:
Status: Undecided
Type: Undergraduate
Department: Computer Science and Engineering
Recommendation: Accept

4. Acceptance Threshold for “undergraduate” and “Computer Science and Engineering”: 10

Expected Results The status of application #1234 is changed to “accepted” from “undecided”. No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “x”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Change the application status from “undecided” to the application recommendation.
4. Verify that the application status is “accepted”.

Owners: Krystal Lee

C2.2 College Level User Change Final Application Status to Waitlisted

Description Ensure that a valid college level user can change the application status of an application from “undecided” to “waitlisted”.

Test Inputs Shared database initialized with the following:
1. College level user account “x”.
2. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Department: Computer Science and Engineering
Recommendation: Waitlist
Score: 85

2. Application #2345 that contains the following fields:
Status: Waitlisted
Type: Undergraduate
Department: Computer Science and Engineering
Score: 86

3. Application #3456 that contains the following fields:
Status: Waitlisted
Type: Undergraduate
Department: Computer Science and Engineering
Score: 84

4. Waiting-list for “undergraduate” applications to “Computer Science and Engineering” with the following fields:
Removal: Enabled
Applications in decreasing order:

1. Application #2345
2. Application #3456

Expected Results The status of application #1234 is changed to “waitlisted” from “undecided”, and it is added to the waitlist for “undergraduate” applications to “Computer Science and Engineering” department in between application #2345 and #3456. No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “x”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Change the application status from “undecided” to the application recommendation.
4. Verify that the application status is “waitlisted”.
5. Request waitlist for “undergraduate” to “Computer Science and Engineering” department from U-ADMT.
6. Verify that the returned waitlist contains 3 (three) applications.
7. Verify that the 3 (three) applications returned are #2345, #1234, and #3456, in that order.
8. Verify that for each field of each returned application the information is correct.

Owners: Krystal Lee

C2.3 College Level User Change Final Application Status to Rejected

Description Ensure that a valid college level user can change the application status of an application from “undecided” to “accepted”.

Test Inputs Shared database initialized with the following:
1. College level user account “x”.
2. Application #1234 that contains the following fields:
Status: Undecided

Type: Undergraduate
Recommendation: Reject

Expected Results The status of application #1234 is changed to “rejected” from “undecided”. No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “x”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Change the application status from “undecided” to the application recommendation.
4. Verify that the application status is “rejected”.

Owners: Krystal Lee

C2.4 College Level User Change Rejected Application Status

Description Ensure that U-ADMT does not allow an application status change in an application with a status of “rejected”.

Test Inputs Shared database initialized with the following:
1. College level user account “x”.
2. Application #1234 that contains the following fields:
Status: Rejected
Type: Undergraduate

Expected Results The intended test operation does not go through, and there is an exception catched

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “x”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Change the application status from “rejected” to “accepted”.
4. Verify that U-ADMT does not allow the status to be changed from “rejected”, and there is an exception catched.

Owners: Krystal Lee and Qi Wu

C2.5 Invalid College Level User Change Final Application Status

Description Ensure that U-ADMT does not allow an application status change to something that is not “undecided”, “accepted”, “rejected”, “waitlisted”, or “confirmed”

Test Inputs Shared database initialized with the following:
1. College level user account “x”.
2. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate

Expected Results The intended test operation does not go through, and there is an exception catched.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “x”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Change the application status from “undecided” to “ok”.
5. Verify that U-ADMT does not allow the status to be changed to “ok”, and there is an exception catched.

Owners: Krystal Lee and Qi Wu

C3.1 College Level User set Response Deadline

Description Ensure that a valid college level user can set the response deadline for an application.

Test Inputs Shared database initialized with the following:
1. College level user account “x”.
2. Application #1234 that contains the following fields:
Status: Accepted
Type: Undergraduate
Deadline: None

Expected Results The Deadline of application #1234 is changed to 11:55p.m., May,14th,2016. No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “x”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Set the response deadline for the application to 11:55p.m., May, 14th, 2016.
4. Verify that the application response deadline for that application in system is set to 11:55p.m., May,14th,2016.

Owners: Qi Wu

C4.1 College Level User to set Requirement Threshold

Description Ensure that a valid college level user can set the requirement threshold.

Test Inputs Shared database initialized with the following:
1. College level user account “x”.
2. Application #1234 that contains the following fields:
Status: Undecide
Type: Undergraduate
Score: 90
Application #1235 that contains the following fields:
Status: Undecide
Type: Undergraduate
Score: 50

Expected Results The application #1235 is auto rejected. No exception is thrown.

Dependencies E5.1, E5.2

Initialization Shared database is loaded, and access to the system has been set through “x”.

Test Steps
1. Set the requirement threshold for score to 60.
2. Verify that application #1235 is auto rejected.

Owners: Qi Wu

C5.1 College Level User to Enable Waitlist

Description Ensure that a valid college level user can enable waitlist removal.

Test Inputs Shared database initialized with the following:
1. College level user account “x”
Waitlist Removal: Disabled.

Expected Results Waitlist removal is enabled. No exception is thrown.

Dependencies E2.3

Initialization Shared database is loaded, and access to the system has been set through “x”.

Test Steps
1. Enable waitlist removal.
5. Verify that the waitlist removal is enabled.

Owners: Qi Wu

C5.2 College Level User Waitlist Multiple Applications

Description Ensure that a valid college level user can change the application status of multiple applications from a department at once.

Test Inputs

Shared database initialized with the following:
1. College level user account “x”.
2. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Department: Computer Science and Engineering
Recommendation: Waitlist
Score: 85

2. Application #2345 that contains the following fields:
Status: Undecided
Type: Undergraduate
Department: Computer Science and Engineering
Recommendation: Waitlist
Score: 86

3. Application #3456 that contains the following fields:
Status: Undecided
Type: Undergraduate
Department: Computer Science and Engineering
Recommendation: Waitlist
Score: 84

4. Waitlist for “undergraduate” applications to “Computer Science and Engineering” with the following fields:
Removal: Enabled
Applications in decreasing order: None

Expected Results

The status of application #1234 is changed to “accepted” from “undecided”. No exception is thrown.

Dependencies

None.

Initialization

Shared database is loaded, and access to the system has been set through “x”.

Test Steps

1. Waitlist bottom 2 (two) applications of type “undergraduate” for department “Computer Science and Engineering”.
2. Request the waitlist of“undergraduate” applications “undergraduate” for department “Computer Science and Engineering” from U-ADMT
3. Verify that the returned list contains 2 (two) applications.
4. Verify that the 2 (two) returned applications are #1234 and #3456, in that order.

5. Verify that for each field of each returned application the information is correct.

Owners: Krystal Lee

C6.1 College Level User Select Top Undergraduate Applications by Score

Description Ensure that a valid college level user can select the desired top number of students for each department and application type based on application scores.

Test Inputs Shared database initialized with the following:

1. College level user account “x” that contains the following field:
Department: Computer Science and Engineering

2 Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Department: Computer Science and Engineering
Score: 99

3 Application #5678 that contains the following fields:
Status: Undecided
Type: Undergraduate
Department: Computer Science and Engineering
Score: 80

4 Application #9012 that contains the following fields:
Status: Undecided
Type: Undergraduate
Department: Computer Science and Engineering
Score: 0

5 Application #9013 that contains the following fields:
Status: Undecided
Type: Graduate
Department: Computer Science and Engineering
Score: 85

6 Application #9014 that contains the following fields:
Status: Undecided
Type: Undergraduate
Department: Chemical Engineering
Score: 100

Expected Results A list of 2 (two) applications in descending order #1234 and #9012. No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through "x".

Test Steps

1. Select the top 2 (two) applications of type "undergraduate" for department "Computer Science and Engineering" from U-ADMT.
2. Verify that the returned list of selected applications contains 2 (two) applications.
3. Verify that the 2 (two) returned applications are #1234 and #5678, in that order.
4. Verify that for each field of each returned application the information is correct.

Owners: Krystal Lee

C7.1 College Level User Change Acceptance Threshold

Description Ensure that a valid college level user can change the acceptance threshold for a department and application type.

Test Inputs Shared database initialized with the following:
1. College level user account "x".
2. Acceptance Threshold for "undergraduate" and "Computer Science and Engineering": 0
3. Acceptance Threshold for "graduate" and "Computer Science and Engineering": 10
4. Acceptance Threshold for "professional" and "Computer Science and Engineering": 100

Expected Results The acceptance threshold for "undergraduate" and "Computer Science and Engineering" is changed to 10 (ten). The acceptance threshold for "graduate" and and "Computer Science and Engineering" is changed to 100 (hundred). The acceptance threshold for "professional" and "Computer Science and Engineering" is changed to 0 (zero). No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through "x".

Test Steps

1. Request the acceptance threshold for "undergraduate" and "Computer Science and Engineering" from U-ADMT.
2. Verify that the acceptance threshold is 0 (zero).
3. Change the acceptance threshold to 10 (ten).
4. Verify that the acceptance threshold is 10 (ten).

5. Request the acceptance threshold for “graduate” and “Computer Science and Engineering” from U-ADMT.
6. Verify that the acceptance threshold is 10 (ten).
7. Change the acceptance threshold to 100 (hundred).
8. Verify that the acceptance threshold is 100 (hundred).
9. Request the acceptance threshold for “professional” and “Computer Science and Engineering” from U-ADMT.
10. Verify that the acceptance threshold is 100 (hundred).
11. Change the acceptance threshold to 0 (zero).
12. Verify that the acceptance threshold is 0 (zero).

Owners: Krystal Lee

C7.2 College Level User Change Final Application Status to Accepted With Met Acceptance Threshold

Description Ensure that if a valid college level user changes the application status of an application from “undecided” to “accepted”, and the acceptance threshold has been met then U-ADMT changes the application status to “waitlisted” instead.

Test Inputs Shared database initialized with the following:
1. College level user account “x”.
2. Application #1234 that contains the following fields:
Status: Undecided
Type: Undergraduate
Department: Computer Science and Engineering
Recommendation: Accept

3. Acceptance Threshold for “undergraduate” and “Computer Science and Engineering”: 0

Expected Results The status of application #1234 is changed to “waitlisted” from “undecided”. No exception is thrown.

Dependencies None.

Initialization Shared database is loaded, and access to the system has been set through “x”.

Test Steps

1. Request application #1234 from U-ADMT.
2. Verify that the information in the returned application is correct.
3. Change the application status from “undecided” to the application recommendation.
4. Verify that the application status is “waitlisted”.

Owners: Krystal Lee