HiLingual Server Test Cases

Team 2

Garrett Davidson, Noah Maxey, Nate Ohlson, Riley Shaw

Validating session token

1. Title: Sending a request with a valid session token
   Procedure: A user sends a POST request to the server containing a valid alphanumeric session token.
   Result: The server responds that the session token is valid.
   Severity: Critical

2. Title: Sending a request with a different user's session token
   Procedure: A user sends a POST request to the server containing an active alphanumeric session token for a different user.
   Result: The server responds that the session token is valid.
   Severity: Critical

3. Title: Sending a request with an invalid session token
   Procedure: A user sends a POST request to the server containing an invalid alphanumeric session token.
   Result: The server responds that the session token is invalid.
   Severity: Critical

4. Title: Sending a request with an improperly formatted session token
   Procedure: A user sends a POST request to the server with a session token containing non-alphanumeric characters.
   Result: The server responds that the session token is invalid.
   Severity: Important

5. Title: Sending a request with an improperly non-ASCII session token
   Procedure: A user sends a POST request to the server with a session token containing non-ASCII characters.
   Result: The server responds that the session token is invalid.
   Severity: Important

6. Title: Sending a request with an empty session token.
   Procedure: A user sends a POST request to the server containing an empty session token.
   Result: The server responds that the session token is invalid.
   Severity: Workaround

7. Title: Sending a request with no session token
   Procedure: A user sends a POST request to the server containing no session token.
   Result: The server responds that the session token is invalid.
   Severity: Workaround

Sending a request to a user id

8. Title: Sending a request with a valid user id
   Procedure: A user sends a POST request to the server containing their valid alphanumeric session token and a valid recipient user id.
   Result: The server responds that the user id is valid.
   Severity: Important

9. Title: Sending a request with a user id for a nonexistent user
   Procedure: A user sends a POST request to the server containing their valid alphanumeric session token and a recipient user id for a nonexistent user.
   Result: The server responds that the user id is invalid.
   Severity: Workaround

10. Title: Sending a request with an invalid id
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token and a non-numeric string for the user id.
    Result: The server responds that the user id is invalid.
    Severity: Workaround

11. Title: Sending a request with an empty recipient
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token and an empty recipient.
    Result: The server responds that the user id is invalid.
    Severity: Workaround

12. Title: Sending a request with no recipient specified
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token and no recipient specified.
    Result: The server responds that the user id is invalid.
    Severity: Workaround

13. Title: Multiple concurrent requests
    Procedure: Simultaneously send several valid requests for each of the supported request types to the server, each on behalf of a different user.
    Result: Each of the requests should be executed properly and succeed.
    Severity: Workaround

Picture messages

14. Title: Sending a picture message
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token, a valid recipient user id, and valid PNG image data.
    Result: The server returns 200.
    Severity: Critical

15. Title: Sending a picture message with extra fields
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token, a valid recipient user id, valid PNG image data, and additional fields with arbitrary data.
    Result: The server ignores the additional fields and returns 200.
    Severity: Workaround

16. Title: Sending a too large picture message
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token, a valid recipient user id, valid PNG image data greater than 10 MB.
    Result: The server responds that the message is too large.
    Severity: Workaround

17. Title: Sending a picture message to yourself
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token, their own user id, and valid PNG image data.
    Result: The server responds that they cannot send a message to themselves.
    Severity: Workaround

18. Title: Sending a picture message with invalid image data
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token, a valid recipient user id, and non-PNG data.
    Result: The server responds that the image is invalid.
    Severity: Important

19. Title: Sending a picture message with empty image data
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token, a valid recipient user id, and an empty value for the image data.
    Result: The server responds that the image is invalid.
    Severity: Workaround

20. Title: Sending a picture message with no image data
    Procedure: A user sends a POST request to the server containing only their valid alphanumeric session token and a valid recipient user id.
    Result: The server responds that the image is invalid.

Severity: Workaround

21. Title: Retrieving your messages from nonexistent user
    Procedure: A user sends a GET request to the server containing a valid session token and a nonexistent user id.
    Result: The server responds that the user is invalid.
    Severity: Workaround

22. Title: Retrieving your messages from a valid user that you have corresponded with
    Procedure: A user sends a GET request to the server containing a valid session token and a valid user id for a user that they have corresponded with.
    Result: The server responds with an array of messages with that user.
    Severity: Important

23. Title: Retrieving your messages from a valid user that you have not corresponded with
    Procedure: A user sends a GET request to the server containing a valid session token and a valid user id for a user that they have not corresponded with.
    Result: The server responds with an empty array.
    Severity: Workaround

Flashcards

24. Title: Sending an array of flashcards to server
    Procedure: A user sends a POST request to the server containing a valid session token and a valid JSON array of flashcards.
    Result: The server responds with the flashcard set ID.
    Severity: Important

25. Title: Sending an invalid array of flashcards to server
    Procedure: A user sends a POST request to the server containing a valid session token and an invalid JSON string.
    Result: The server responds that the flashcards are invalid.
    Severity: Workaround

26. Title: Receiving flashcards from server
    Procedure: A user sends a GET request to the server containing a valid session token and a valid flashcard set ID.
    Result: The server responds with an array containing all of the flashcards from that set.
    Severity: Important

27. Title: Requesting nonexistent flashcards from server
    Procedure: A user sends a GET request to the server containing a valid session token and an ID for a nonexistent flashcard set.
    Result: The server responds with an empty array.

Severity: Workaround

28. Title: Sending empty flashcard front
    Procedure: A user sends a POST request to the server containing a valid session token and an empty string value for the flashcard front and a valid string value for the flashcard back.
    Result: The server responds that the flashcard is invalid.
    Severity: Workaround

29. Title: Sending empty flashcard back
    Procedure: A user sends a POST request to the server containing a valid session token and an empty string value for the flashcard back and a valid string value for the flashcard front.
    Result: The server responds that the flashcard is invalid.
    Severity: Workaround

30. Title: Omitting the flashcard front
    Procedure: A user sends a POST request to the server containing only a valid session token and valid string for flashcard back.
    Result: The server responds that the flashcard is invalid.
    Severity: Workaround

31. Title: Omitting flashcard back
    Procedure: A user sends a POST request to the server containing only a valid session token and valid string for flashcard front.
    Result: The server responds that the flashcard is invalid.
    Severity: Workaround

32. Title: Sending too long flashcard front
    Procedure: A user sends a POST request to the server containing a valid session token, a valid string for flashcard back, and a string of more than 50 Unicode characters for the flashcard front.
    Result: The server responds that the flashcard is invalid.
    Severity: Important

33. Title: Sending too long flashcard back
    Procedure: A user sends a POST request to the server containing a valid session token, a valid string for flashcard front, and a string of more than 50 Unicode characters for the flashcard back.
    Result: The server responds that the flashcard is invalid.
    Severity: Important

Profile

34. Title: Receiving profile from the server
    Procedure: A user sends a GET request to the server containing a valid session token.
    Result: The server responds with an object containing all the fields that make up the user's profile.
    Severity: Important

35. Title: Sending an edited profile to the server
    Procedure: A user sends a POST request to the server containing a valid session token and valid object with the user's profile information.
    Result: The server responds that the profile has been updated
    Severity: Important

36. Title: Sending an invalid edited profile to the server
    Procedure: A user sends a POST request to the server containing a valid session token and invalid object with the user's profile information.
    Result: The server responds that the profile edit is invalid.
    Severity: Important

37. Title: Sending a profile to the server with an invalid age
    Procedure: A user sends a POST request to the server containing a valid session token and object with the user's profile information and an age with non-numeric value.
    Result: The server responds that the profile age is invalid.
    Severity: Workaround

38. Title: Sending a profile to the server with an invalid age
    Procedure: A user sends a POST request to the server containing a valid session token and object with the user's profile information and an age less than 0.
    Result: The server responds that the profile age is invalid
    Severity: Workaround

39. Title: Sending a profile to the server with an empty age
    Procedure: A user sends a POST request to the server containing a valid session token and object with the user's profile information and an empty age.
    Result: The server responds that the profile age is empty.
    Severity: Workaround

40. Title: Sending a profile to the server with no age
    Procedure: A user sends a POST request to the server containing a valid session token and object with the user's profile information and no age.
    Result: The server responds that the profile has no age.
    Severity: Workaround

41. Title: Sending a profile to the server with an empty name
    Procedure: A user sends a POST request to the server containing a valid session token and object with the user’s profile information and an empty name.
    Result: The server responds that the profile name is empty.
    Severity: Workaround

42. Title: Sending a profile to the server with no name
    Procedure: A user sends a POST request to the server containing a valid session token and object with the user’s profile information and no name.
    Result: The server responds that the profile has no name.
    Severity: Workaround

43. Title: Sending a profile to the server with a name > 100 chars
    Procedure: A user sends a POST request to the server containing a valid session token and object with the user’s profile information and a name > 100 chars.
    Result: The server responds that the profile’s name is too long.
    Severity: Workaround

44. Title: Sending a profile to the server with an invalid language
    Procedure: A user sends a POST request to the server containing a valid session token and object with the user’s profile information and language that doesn’t match one of our supported languages.
    Result: The server responds that the profile language is invalid.
    Severity: Workaround

45. Title: Sending a profile to the server with multiple languages
    Procedure: A user sends a POST request to the server containing a valid session token and object with the user’s profile information including a list of languages.
    Result: The server responds that the profile has been updated.
    Severity: Workaround

46. Title: Sending a profile to the server with an empty language
    Procedure: A user sends a POST request to the server containing a valid session token and object with the user’s profile information and an empty language.
    Result: The server responds that the profile language is empty.
    Severity: Workaround

47. Title: Sending a profile to the server with no language
    Procedure: A user sends a POST request to the server containing a valid session token and object with the user’s profile information and no language field.
    Result: The server responds that the profile has no language.
    Severity: Workaround

48. Title: Sending invalid profile edit
    Procedure: A user sends a POST request to the server with a valid session token and an invalid object with the user's profile information.
    Result: The server responds that it was a bad request.
    Severity: Workaround

49. Title: Sending a new profile to the server
    Procedure: A user sends a POST request to the server containing a new auth token from Facebook or Google and valid object with the user's profile information.
    Result: The server responds with the profile's new user id.
    Severity: Important

50. Title: Sending an invalid new profile to the server
    Procedure: A user sends a POST request to the server containing a new auth token from Facebook or Google and invalid object with the user's profile information.
    Result: The server responds that the profile was invalid.
    Severity: Important

51. Title: Creating a duplicate account
    Procedure: A user sends a POST request to the server containing an auth token from Facebook or Google for a Facebook or Google account that is already associated with an existing account and valid object with the user's profile information.
    Result: The server responds that the profile has already been created.
    Severity: Important

<u>Push Notifications</u>

52. Title: Receiving a push notification
    Procedure: User A sends a POST request to the server containing a message, the id of User B as the recipient, and a valid session token.
    Result: User B should receive a notification.
    Severity: Workaround

53. Title: Receiving a push notification
    Procedure: User A sends a POST request to the server containing a message, an invalid recipient user id, and a valid session token.
    Result: The server responds that the recipient is invalid.
    Severity: Workaround

54. Title: Receiving a push notification
    Procedure: User A sends a POST request to the server containing a message, the id of User B as the recipient, and an invalid session token.
    Result: The server responds that the session token is invalid.
    Severity: Workaround

Audio Message

55. Title: Sending an audio message
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token, a valid recipient user id, and valid m4a audio data.
    Result: The server returns 200.
    Severity: Important

56. Title: Sending an audio message with extra fields
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token, a valid recipient user id, valid audio data, and additional fields with arbitrary data.
    Result: The server ignores the additional fields and returns 200.
    Severity: Workaround

57. Title: Sending too large audio message
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token, a valid recipient user id, valid audio data greater than 10 MB.
    Result: The server responds that the file is too large.
    Severity: Workaround

58. Title: Sending an audio message to yourself
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token, their own user id, and valid audio data.
    Result: The server responds that they cannot send a message to themselves.
    Severity: Workaround

59. Title: Sending an audio message with invalid audio data
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token, a valid recipient user id, and invalid m4a data.
    Result: The server responds that the audio is invalid.
    Severity: Workaround

60. Title: Sending an audio message with empty audio data
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token, a valid recipient user id, and an empty value for the audio data.
    Result: The server responds that the audio is invalid.
    Severity: Workaround

61. Title: Sending an audio message with no audio data
    Procedure: A user sends a POST request to the server containing only their valid alphanumeric session token and a valid recipient user id.
    Result: The server responds that the audio is invalid.
    Severity: Workaround

Text Message

62. Title: Sending a text message
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token, a valid recipient user id, and a non-empty message string.
    Result: The server returns 200.
    Severity: Critical

63. Title: Sending a too large text message
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token, a valid recipient user id, and a non-empty message string containing more than 500 Unicode characters.
    Result: The server responds that the message is too long.
    Severity: Critical

64. Title: Sending a text message with an invalid session token
    Procedure: A user sends a POST request to the server containing an invalid session token, a valid recipient user id, and a non-empty string.
    Result: The server responds that the session token is invalid.
    Severity: Important

65. Title: Sending a text message with an invalid recipient
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token, an invalid recipient user id, and a non-empty string.
    Result: The server responds that the recipient is invalid.
    Severity: workaround

66. Title: Sending a text message with an empty message
    Procedure: A user sends a POST request to the server containing their valid alphanumeric session token, an invalid recipient user id, and an empty string.
    Result: The server responds that the message must not be empty.
    Severity: workaround

67. Title: Sending a text message with no message
    Procedure: A user sends a POST request to the server containing only their valid alphanumeric session token and a valid recipient user id.
    Result: The server returns 200.
    Severity: Workaround

Translations

68. Title: Sending a translation request
    Procedure: A user sends a POST request to the server containing their valid session token, the requested phrase to be translated <= 500 chars, and the language the requested phrase should be translated to.
    Result: The server returns the translated string.
    Severity: Critical

69. Title: Sending a translation request with a requested phrase translation of size greater than 500 characters
    Procedure: A user sends a POST request to the server containing a valid session token, a string longer than 500 characters to be translated, and the language the requested phrase should be translated to.
    Result: The server responds that the requested phrase is too long.
    Severity: Workaround

70. Title: Sending a translation request with an empty requested translation phrase
    Procedure: A user sends a POST request to the server containing a valid session token, an empty requested translation phrase, and the language the requested phrase should be translated to.
    Result: The server responds that the requested phrase is empty.
    Severity: Workaround

71. Title: Sending a translation request with a requested phrase translation that is in multiple languages
    Procedure: A user sends a POST request to the server containing a valid session token, a requested phrase translation that is in multiple languages, and the language the requested phrase should be translated to.
    Result: The server responds that the requested phrase is not valid.
    Severity: Workaround

72. Title: Sending a translation request with a requested phrase translation with multiple alphabets.
    Procedure: A user sends a POST request to the server containing a valid session token, a requested phrase to be translated written with multiple alphabets (e.g. a mix of English and Arabic text), and the language the requested phrase should be translated to.
    Result: The server responds that the requested phrase is not valid.
    Severity: Workaround

73. Title: Sending a translation request with a target language that does not match our supported languages.
    Procedure: A user sends a POST request to the server containing a valid session token,

a requested phrase to be translated, and a target language that does not match one of our supported languages.
Result: The server responds that the requested target language is not supported.
Severity: Workaround

Matching

74. Title: Sending a matching request and receiving matches
Procedure: Populate the database with users that know a language the requester is learning and vice versa. The requester sends a GET request to receive their match list.
Result: The server returns a non-empty array of matching profiles, each of whom is learning a language that the requesting user knows and knows a language the requesting user is learning.
Severity: Important

75. Title: No valid matches
Procedure: Populate the database exclusively with users who are learning a language the requester knows but do not know a language the requester is learning, users who know a language the requester is learning but are not learning a language the requester knows, and users who neither know a language the requester is learning nor are learning a language the requester knows. The requester sends a GET request to receive their match list.
Result: The server returns an empty array.
Severity: Important

76. Title: Sending a matching request and receiving more than 10 users
Procedure: Populate the database with more than 10 users that match the requester's language selection. A user sends a GET request to receive their match list.
Result: The server returns an array of no more than 10 matching profiles.
Severity: Workaround

Authorization

77. Title: Sending an authorization token to sign in via Facebook/Google
Procedure: A user sends a POST request to the server containing a valid Facebook/Google id, and the corresponding Facebook/Google authorization token.
Result: The server returns the requestor's HiLingual session token.
Severity: Important

78. Title: Sending an authorization to sign in via Facebook/Google with invalid id
Procedure: A user sends a POST request to the server containing an invalid Facebook/Google id, and a valid Facebook/Google authorization token.
Result: The server responds that the authorization failed.
Severity: Important

79. Title: Sending an authorization to sign in via Facebook/Google with invalid token
    Procedure: A user sends a POST request to the server containing a valid Facebook/Google id, and an invalid Facebook/Google authorization token.
    Result: The server responds that the authorization failed.
    Severity: Important