

System and Unit Test Report – Voice Capsule by the Voice Capsulators, Revision Date 11/21/2021

SPRINT 1:

- A. User Story 1 from Sprint 1: *As someone who values privacy, I want to be able to use an app which lets me log in and out of an account securely.*
- B. User Story 2 from Sprint 1: *As a database engineer, I value systems that handle data safely.* **[This user story has no current functionality in the Mobile Application; it's purpose was for research. See Sprint Plan 1 for further details.]**

Scenario 1: Signing up

1. Open Voice Capsule app
2. Press 'SIGN-UP' button
3. User should be directed to 'Create Account' page
4. User should be able to input following information into the three fields which appear:
 - a. Email = <'signuptester@gmail.com'>
 - b. Name = <'Signup Tester'>
 - c. Password = <'password'>
5. Click 'SAVE'
6. This should open the 'Record' Page. User has successfully logged in

Scenario 2: Signing up with invalid credentials - already existing log in

1. Follow steps 1-5 from scenario 1, although steps 4b and 4c can be a different name and password
2. Should see pop-up "Failed to create an account - The email address is already in use by another account."

Scenario 3: Signing up with invalid credentials - invalid email address

1. Follow steps 1-3 from scenario 1
2. User should input an invalidly formatted email into the email section, and fill name and password fields. For example:
 - a. Email = <hahfdahoeiaf">
 - b. Name = <'Signup Tester>
 - c. Password = <'password'>
3. Click 'SAVE'
4. Pop-up should appear saying "Failed to create account - The email address is badly formatted"

Scenario 4: Signing up with invalid credentials - weak password

5. Follow steps 1-3 from scenario 1
6. User should input an email, name and password less than 6 characters into the fields that appear. For example:
 - a. Email = <signup2@gmail.com">
 - b. Name = <'Signup Tester>
 - c. Password = <'haha'>
7. Click 'SAVE'
8. Pop-up should appear saying "Password should be at least 6 characters"

Scenario 5: Logging in with valid credentials

1. Open Voice Capsule App
2. Press 'LOGIN' button
3. User should be directed to 'Sign in with email' page

4. User should input the following into the 'Enter your email' field:
 - a. Email = <'signuptester@gmail.com">
5. Press 'NEXT' button
6. User should be directed to 'Sign in' page
7. User should input the following into the 'Password' field:
 - a. Password = <password>
8. Press 'SIGN-IN' button
9. User should be successfully logged in and directed to the record page

Scenario 6: Logging in with invalid credentials - email does not belong to a user

1. Follow steps 1-3 from Scenario 5
2. User should input an email which does not belong to an account into the 'Enter your email' field. For example:
 - a. Email = <testtesttest@gmail.com">
3. Pop-up should appear saying "Invalid email - This email does not currently belong to a user account".

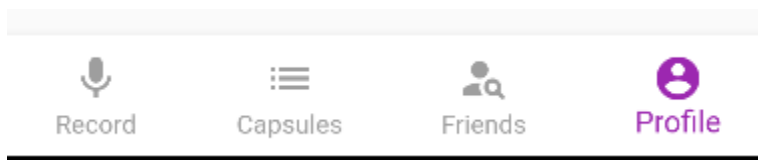
Scenario 7: Logging in with invalid credentials - email is badly formatted (maybe user made a typo)

1. Follow steps 1-6 from Scenario 5
2. User should input an incorrect password to the following into the 'Password' field. For example:
 - a. Password = <password1>]
3. Press Sign-in button

4. Pop-up should appear saying “Failed to sign in - The password is invalid or the user does not have a password.”

Scenario 8: Logging out

1. User should carry out steps 1-6 from Scenario 1
2. User should tap on the Profile button on the far right at the navigation bar on the bottom of the screen:




3. User should be directed to the Profile page
4. User should be press the ‘LOGOUT’ button that is just below the email field
5. User should be directed back to the ‘Login to Voice Capsule’ page

SPRINT 2:

- A. User Story 1 from Sprint 2: *As a mindful individual, I want to send my future-self positive messages and affirmations.*
- B. User Story 2 from Sprint 2: *As someone who loves hearing messages from the past, I want to be able to download and save voice notes that have been sent to me. [This particular user story was not completed until Sprint 4, refer to Sprint 4 for the scenario]*

Scenario 1: Recording functionality test

- 1) Start on the “recording” page
 - a) Click circular red button () to start recording

- 2) Button icon should switch to “stop” button icon to indicate recording has started
- 3) Text indicator under the recording button should switch to say “recording” to indicate that the app is currently recording audio
- 4) User should see a vertical bar-style audio visualizer above the recording button indicating the level of audio that is being recorded
 - a) Click the square red button (■) to stop recording
- 5) User should see the recording button revert to a circle; the text indicator should return to the “record” prompt; the audio input visualizer should disappear
- 6) The app should transition to the “send” screen where the user can review their recording with the player module

Scenario 2: Playback module functionality test on “send” screen

- 1) Start on the “send” page directly after recording a Voice Capsule
 - a) Click the triangle play button (▶) on the player interface
- 2) Playback module should start playing, play button should change to pause button (⏸), and user should be able to hear the recording
 - a) Click the counter-clockwise circular arrow button (↶) to skip backwards 5 seconds
- 3) Playback module should skip backwards 5 seconds and continue playing
 - a) Click the clockwise circular arrow button (↷) to skip forward 5 seconds
- 4) Playback module should skip forwards 5 seconds and continue playing
 - a) Click the double left-pointing arrows button (⏮) to restart playback
- 5) Playback module should restart and continue playing
 - a) Click the pause button (⏸)
- 6) Playback module should pause at the current timestamp

- a) Click the square stop button (■)
- 7) Playback module should return to the beginning of the recording
 - a) Click the clockwise circular arrow button (↻) to skip forward 5 seconds
- 8) Playback module should skip forwards 5 seconds without starting to play
 - a) Click the counter-clockwise circular arrow button (↺) to skip backwards 5 seconds
- 9) Playback module should skip backwards 5 seconds without starting to play
 - a) Click the triangle play button (▶)
- 10) Playback module should start playing
 - a) Wait until the recording is over
- 11) Playback should stop at the end of the recording

Scenario 3: Playback module testing on the “capsules” page


- 1) Start on the “capsules” page with a capsule available to open
 - a) Click on the capsule
- 2) The app should transition to the playback screen and start playing automatically
 - a) Perform the testing sequence outlined in Scenario 2 to determine if the player is functioning correctly

SPRINT 3:

- A. User Story 1 from Sprint 3: *As someone who loves hearing messages from the past, I want to be able to download and save voice notes that have been sent to me.* **[This particular user story was not completed until Sprint 4, refer to Sprint 4 for the scenario]**

- B. User Story 2 from Sprint 3: *As a sociable person, I want to be able to search for and add other users as 'friends' on the app easily***[This particular user story was not completed until Sprint 4, refer to Sprint 4 for the scenario]**
- C. User Story 3 from Sprint 3: *As a forgetful person, I want to send my friend a happy birthday message eight months in advance, but they can only open it on their birthday*

Scenario 3: Sending a capsule to a friend.

1. Start Voice Capsule and select "Login".
2. Login as:
 - a. Email = <'friends_test_user_one@gmail.com'>.
 - b. Password = <'password'>.
3. Click circular red button '' to start recording.
 - a. Ensure that your Android Virtual Device has its microphone turned on.
4. Click the square red button '■' to stop recording.
5. Select an open date in the past.
 - a. This is so we can immediately verify that the capsule was sent.
6. Select a recipient that is not yourself.
 - a. Selected name = <'User Two'>.
 - b. This should show all of the your friends in addition to yourself.
7. Press 'SEND'.
8. Press the 'Profile' tab.
9. Press 'LOGOUT'.
10. Press 'Login' and login as
 - a. Email = <'friends_test_user_two@gmail.com'>.
 - b. Password = <'password'>.
11. Navigate to the 'Capsules' tab.

12. Press 'Refresh'.
13. Open the new capsule.
 - a. You should hear the capsule that 'User One' sent.

SPRINT 4:

- A. User Story 1 from Sprint 4: *As a technophobe, I want to be able to use an app that is simple to navigate and use.*
- B. User Story 2 from Sprint 4: *As someone who loves hearing messages from the past, I want to be able to download and save and voice notes that have been sent to me.*
- C. User Story 3 from Sprint 4: *As a sociable person, I want to be able to search for and add other users as 'friends' on the app easily*

Scenario 1: As a technophobe, I want to be able to use an app that is simple to navigate and use.

- 1) Drop down the notification shade in Android by sliding down from the notification bar
- 2) Toggle Dark Mode using the quick setting toggle
- 3) Log in to Voice Capsule with the following parameters
 - a) Email = <'friends_test_user_two@gmail.com'>.
 - b) Password = <'password'>.
- 4) Record a voice clip by pressing the center button
- 5) Choose the current date on the "Send Recording" screen
- 6) Choose the recipient as "Myself"
- 7) Press the "Send" button
- 8) Press the other buttons in the bottom navigation bar to view previous messages, friends, and one's profile page.

Scenario 2: Completing a recording, setting the date and time for which another user (or one's self) will be able to open the recording, and sending the recording off to Firebase Storage

- 1) Log in to Voice Capsule with the following parameters:
 - a) Email = <'friends_test_user_two@gmail.com'>.
 - b) Password = <'password'>.
- 2) Start on the "Record" tab
 - a) Press the "Record" button to begin recording a voice clip
 - b) Press the button again to stop recording
- 3) A popup will appear stating that the recording has been finished
- 4) Another popup will appear stating that the recording has been saved in cache as "recorded_file.mp4"
- 5) On the "Send Recording" page, a user can choose a particular date and time for the recipient to download and play the recording back
- 6) While on this same page, a user can then choose from a list of added friends or one's self (listed as "Myself") as a recipient of the Voice Capsule
- 7) A user hits the "Send" button below
 - a) If a user has not chosen a selected date, a dialog box will appear and inform the user that a date and time is required for the Voice Capsule to be sent
 - i) Hitting "Ok" after this will go back to the "Send Recording" page
- 8) The file holding the recording will be renamed and sent out with the following format:
 - a) "outgoing_<user ID>_<sent time>" (e.g. "outgoing_BP3BkDO6W2bo30p30sSYqXzLYf32_2021-11-28_20-53-08-490017.mp4" such that "BP3BkDO6W2bo30p30sSYqXzLYf32" is the user ID and "2021-11-28_20-53-08-490017" is the sending date"
- 9) The user will be returned to the "Record" page with a popup stating that the Voice Capsule was successfully sent

Scenario 3: Checking for openable capsules in Firestore Database and downloading audio files from Firebase Storage

- 1) Start on “capsules” page
 - a) Click the “refresh” button on the bottom of the page
- 2) Query to Firestore database checks “pending_capsules” document for capsules whose open_date_time field is prior to the current date/time;
 - a) If openable capsules exist, download them from Firebase Storage;
 - i) User should see a pop-up message saying that new capsules have been received and they should be able to see them in the “capsules” page list as new-marked capsules
 - b) If no openable capsules exist;
 - i) User should see a pop-up message saying that no new capsules are available

Scenario 2 : Requesting a friend

1. Start Voice Capsule and select “SIGN-UP”.
2. Register a new user:
 - a. Email = <'friends_test_user_one@gmail.com'>
 - b. Name = <'User One'>
 - c. Password = <'password'>
3. Navigate to the ‘Friends’ tab.
 - a. It should be empty, and if refresh is pressed, it should remain empty.
4. Press ‘Friend Requests’.
 - a. There should be no friend requests.
5. Return to ‘Friends’ using the back arrow
6. Click ‘Add Friends’.

7. Type a friend's email address.
 - a. The text field should not recognize an improperly formatted email address.
 - i. Type email = <">
 - ii. Once requested, you should be prompted to enter an email address.
 - iii. Type email = <'email'>
 - iv. Once requested, you should be prompted to add a properly formatted email and the request will not go through.
 - v. Type email = <'email@'>
 - vi. Once requested, you should be prompted to add a properly formatted email and the request will not go through.
 - b. You should not be able to add someone who doesn't exist.
 - i. Type email = <'this_user_does_not_exist@gmail.com'>
 - ii. Once requested, you should be prompted to add someone who exists and the request will not go through.
 - c. You should not be able to add yourself.
 - i. Type email = <'friends_test_user_one@gmail.com'>
 - ii. Once requested, you should be prompted to add someone who is not you and the request will not go through.
8. Type email = <'friends_test_user_two@gmail.com'>.
9. Request to be this email's friend.
 - a. The page should navigate to the 'Friends' screen and a 'Toast' should confirm that a request has been sent.
10. Press the 'Profile' tab.
11. Press 'LOGOUT'.
12. Press 'Login' and login as
 - a. Email = <'friends_test_user_two@gmail.com'>.

b. Password = <'password'>.

13. Navigate to the 'Friends' tab.

a. You should get a 'Toast' notification that you have pending friend requests.

b. Contacts = {

'riagonza@ucsc.edu' : 'Rick',

'ricardo_burner_email@gmail.com' : 'Ricardo',

'ricardo_spam_email@gmail.com' : 'Richard'

}

14. Click 'Add Friends'.

a. You should not be able to request someone who is already a friend.

i. Type email = <'riagonza@ucsc.edu">

ii. Once requested, you should be prompted to add someone who is not a friend and the request will not go through.

b. You should not be able to request someone who has sent you a friend request that is currently pending.

i. Type email = <'friends_test_user_one@gmail.com'>

ii. Once requested, you should be told that this user has sent you a friend request and the request will not go through.

Scenario 4: Accepting a friend request

1. Start Voice Capsule and select "Login".

2. Login as:

- a. Email = <'friends_test_user_two@gmail.com'>.
 - b. Password = <'password'>.
3. Navigate to the 'Friends' tab.
 - a. You should get a 'Toast' notification that you have pending friend requests.
 - b. Contacts = {

 'riagonza@ucsc.edu' : 'Rick',

 'ricardo_burner_email@gmail.com' : 'Ricardo',

 'ricardo_spam_email@gmail.com' : 'Richard'

 }
4. Press 'Friend Requests'.
 - a. Requests = {

 'friends_test_user_one@gmail.com' : 'User One',

 'friends_test_random_email@gmail.com' : 'Deny Request'

 }
5. Press the 'X' on 'friends_test_random_email@gmail.com'.
 - a. This denies the request and it should disappear.
 - b. You should get a 'Toast' notification confirming that you denied the request.
6. Press the check mark on 'friends_test_user_one@gmail.com'.
 - a. This accepts the request and it should disappear.
 - b. You should get a 'Toast' notification confirming that you accepted the request.
7. Press the back button to return to 'Friends'.

8. Press 'Refresh'.
 - a. The new friend should appear.
9. Press the 'Profile' tab.
10. Press 'LOGOUT'.
11. Press 'Login' and login as:
 - a. Email = <'friends_test_user_one@gmail.com'>.
 - b. Password = <'password'>.
12. Navigate to the 'Friends' tab.
 - a. Contacts = {

'friends_test_user_two@gmail.com' : 'User Two'

}