TraceAlert - Testing Report

Team 27

Database interaction functions

Test description	Input	Expected output	Current output
Test “createNewUser(user)” from “client/www/js/firebase.js” against different parameters.	Passing no parameters	No new account and document created, message “Error occurred when creating the account: no parameter is passed” should be printed in the console	Same as expected
	Passing wrong types of parameters	No new account and document created, message “Error occurred when creating the account: wrong type of parameter” should be printed in the console	Same as expected
	“user” object passed is lack of important properties	No new account and document created, message “Error occurred when creating the account - insufficient information in the object” should be printed in the console	Same as expected
Test “login (email, password)” from “client/www/js/firebase.js” against different parameters.	Passing incorrect email address or password	Message “Error occurred when logging in” should be printed in the console	Same as expected
	Passing no argument into the function	Message “Error occurred when logging in” should be printed in the console	Same as expected
	Passing only one argument (email address) into the function	Message “Error occurred when logging in” should be printed in the console	Same as expected
	Passing arguments of the wrong type into the function	Message “Error occurred when logging in” should be printed in the console	Same as expected
Test “setRiskyWeight(userId, weight)” from “server/index.js” against different parameters.	Passing correct parameters into the function	The weight number of the user with “userId” in the document should be correctly set to the “weight” passed to the function; message “riskyWeight of user <userId> successfully updated!” should be printed in the console with “userId” being the id passed to the function.	Same as expected

	Passing no parameters into the function	No changes to the database should be made, “Error occurred when setting risky weight: userId was not passed” should be printed in the console	Same as expected
	Passing only one parameter into the function	No changes to the database should be made. “Error occurred when setting risky weight: weight was not passed” should be printed in the console	Same as expected
	Passing wrong types of parameters into the function	No changes to the database should be made. “Error occurred when setting risky weight: wrong types of parameters” should be printed in the console.	Same as expected
	Passing non-existing user ID into the function	No changes to the database should be made. “Error occurred when updating user's riskyWeight: Error: 5 NOT_FOUND: No document to update: projects/tracealert-94669/databases/(default)/documents/users/<invalid id>” should be printed in the console with “invalid id” being the incorrect user ID passed to the function	Same as expected
Test “getRiskyWeight (userId)” from “server/index.js” against different parameters.	Passing correct ID into the function	Correct risky weight value should be returned	Same as expected
	Passing non-existing user ID into the function	Message “This user does not exist” should be printed in the console	Same as expected
	Passing no parameters into the function	Message “userId cannot be null” should be printed in the console	Same as expected
	Passing argument of the wrong type into the function	Message “userId is of invalid type” should be printed in the console	Same as expected
Test “updateRiskStatus (userId, atRisk)” from “server/index.js” against different parameters.	Passing correct parameters into the function	The risk status of the user should be correctly updated	Same as expected
	Passing non-existing userId into the function	Message “Error occurred when updating user's risk status: Error: 5 NOT_FOUND: No document to update: projects/tracealert-94669/databases/(default)/documents/users/<invalid id>” should be printed in the console.	Same as expected
	Passing no parameters into the function	Message “Error occurred when updating risk status: the userId is not specified” should be printed in the console.	Same as expected
	Passing only one parameter into the function	Message “Error occurred when updating risk status: atRisk is not specified” should be printed in the console.	Same as expected
	Passing arguments of the wrong types	Message “Error occurred when updating risk status: arguments are of the wrong types” should be printed in the console.	Same as expected

Test “getRiskStatus (userId)” from “server/index.js” against different parameters.	Passing correct ID into the function	The risk status of the user should be correctly updated	Same as expected
	Passing non-existing user ID into the function	Message “Error occurred when getting risk status: TypeError: Cannot read property 'atRisk' of undefined” should be printed in the console.	Same as expected
	Passing no parameters into the function	Message “Error occurred when updating risk status: userId is not provided” should be printed in the console	Same as expected
	Passing argument of the wrong type into the function	Message “Error occurred when updating risk status: userId is of the wrong type” should be printed in the console	Same as expected
Test “updateRiskParameters(riskLevel, threshold, tiers)” from “server/index.js” against different parameters.	Passing correct parameters into the function	All parameters should be correctly updated; message “Risk parameters successfully updated!” should be printed in the console	Same as expected
	Function called with only one/two parameters which are all in correct format	If there is only one parameter, “Risk level successfully updated!” should be printed; if there are two parameters passed, “Risk level and threshold successfully updated!” should be printed in the console.	Same as expected
	Passing no parameters	Message “No parameter specified” should be printed	Same as expected
	Passing arguments of wrong types	Message “Error occurred when updating risk parameters: wrong types of arguments being passed” should be printed	Same as expected
Test “getDirectContacts(userId)” from “server/index.js” against different parameters.	Passing a correct userId	A list of IDs of the direct contacts should be returned	Same as expected
	Passing non-existing user ID into the function	Message “Error occurred when getting direct contacts: the userId does not exist” should be printed in the console.	Same as expected
	Passing no parameters into the function	Message “Error occurred when getting direct contacts: no userId provided” should be printed in the console.	Same as expected
Test type: unit testing Date: 2021-5-31		Responsible: Yumeng Zhang Reviewer: Kaiyang Xue, Xilin Huang, Xuesong Ji	

Test description	Input	Expected output	Current output
Test “createNewUser(user)” from “client/www/js/firebase.js” against different parameters.	The parameter “user” being in its correct format	New account created under “Authentication” in Firebase; New document for the user created under “users” collection in Firestore with personal information filled in.	Same as expected

Test “login(email, password)” from “client/www/js/firebase.js” against different parameters.	Correct email and password	Messages “Signed in!” and “User sign-in state is persistent from now on” should be printed in the console	Same as expected
Test “addNewContact (contact)” from “client/www/js/firebase.js” against different parameters	Passing “contact” in its correct format into the function	Both “contactCount” and “placesAndContacts” subcollection of the user should be correctly updated	Same as expected
Test “getUserInfo()” from “client/www/js/firebase.js” after integration with the user interface.	N/A	Information including first name, last name, date of birth, phone number, email address, address, city, state and postal code should be correctly presented.	Same as expected
Test “getContacts()” from “client/www/js/firebase.js” after integration with the user interface.	N/A	A list of maps should be returned. Each map should store the location information and a list of direct contact records. Each direct contact record should include the ID of the contact and a timestamp. This information should be presented on the screen.	Same as expected
Test “deleteAllOutdatedContacts()” from “server/index.js”	N/A	All the direct contacts recorded outside of the last 14 days should be deleted from the database.	Same as expected
Test type: integration testing Date: 2021-5-31		Responsible: Yumeng Zhang Reviewer: Kaiyang Xue, Xilin Huang, Xuesong Ji	

Input areas in the user interface

Test description	Input	Expected output	Current output
Test “Sign Up” function	Click on “Sign Up” button	The “SignUp” page should be shown, the user can type in his details including phone number, email, password, name(first name and last name), birthday, address, city, state and postal code. All of the information should be filled, otherwise “Please enter” should be displayed.	Same as expected

Test “Sign In” function	Click on “Sign In” button	The “Sign In” page should be shown, the user can type in his account information that he signed up before. Once the user logged in, the page should jump to the user Info page and display “Sign in success!” The user details should be the same as the user type in while signing up.	Same as expected
Test “dateOfBirth” function	N/A	date of birth should be shown as Week/Month/Day/Year, Time should not be displayed.	Same as expected
Test type: scenario testing Date: 2021-5-31		Responsible: Kaiyang Xue Reviewer: Xilin Huang, Xuesong Ji, Yumeng Zhang	

Access to the location information

Test description	Input	Expected output	Current output
Test “location()” from “client/www/js/bluetoothlocation.js” for geolocating function.	N/A	The plugin function “navigator.geolocation.getCurrentPosition()” should receive the user's current location by GPS, pass the longitude and latitude to another plugin function “nativegeocoder.reverseGeocode()” for further processing.	Same as expected.
Test “onSuccess()” from “client/www/js/bluetoothlocation.js” for geocoding function.	N/A	The plugin function “nativegeocoder.reverseGeocode()” should generate the address with longitude and latitude that is received by “navigator.geolocation.getCurrentPosition()”.	Same as expected.
Test type: Integration testing Date: 2021-5-31		Responsible: Xilin Huang Reviewer: Yumeng Zhang, Kaiyang Xue, Xuesong Ji	

Advertisement of ID and detection of direct contacts by the use of Bluetooth

Test description	Input	Expected output	Current output

Test “startAdvertising()” from “client/www/js/bluetoothlocation.js” for Bluetooth advertising.	N/A	The device should keep advertising the message with the user's ID and our customized identity code by the plugin function “bluetoothle.startAdvertising()”. The advertising function should run for 19.5 minutes, and then go into the scanning function.	Same as expected.
Test “startScan()” from “client/www/js/bluetoothlocation.js” for Bluetooth advertisement scanning.	N/A	The device should start scanning for Bluetooth advertisements from other devices that have RSSI less than -80, which means that device is about 6 feet or less. Once it finds the messages with our customized identify code, the function should record the contactor ID and store it into the object. The function should run once every 20 minutes, according to the WHO standard of close contact.	Same as expected.
Test “startScanSuccess()” from “client/www/js/bluetoothlocation.js” for new contact object element inserting.	N/A	After every round of scanning, this function should insert the scanning result and current address into one object. The function should send the object pack with location and multiple contact ID to the database only when the bluetooth scanner gets any result. Dulicaped results in one scan should not be recorded.	Same as expected.
Test type: Integration testing Date: 2021-5-31		Responsible: Xilin Huang Reviewer: Yumeng Zhang, Kaiyang Xue, Xuesong Ji	

Information displayed in the application

Test description	Input	Expected output	Current output
Test the display on different screen sizes	N/A	User interface should be displayed properly on different screen sizes ,especially the size and format of texts, the size and position of pictures.	Same as expected
Test the qr/index	N/A	After the user logs in, The automatically generated qr code can be displayed on the “My Qr Code”page. User name and last updated date should be displayed as well.	Same as expected
Test type: scenario testing Date: 2021-5-31		Responsible: Kaiyang Xue Reviewer: Xilin Huang, Xuesong Ji, Yumeng Zhang	

Table for testing the risk determination and propagation algorithms

Test description	Input	Expected output	Current output
Test the ‘setCoefficients ()’ function from “server/index.js”	N/A	No error in each risk level, and message ‘Coefficients have been set to ‘’ by risk level 0/1//2/3’ should be printed in the console	Same as expected
Test the ‘determineUserRisk (userId)’ function from “server/index.js”	N/A	No error in each risk level, and message ‘The risk status of user ‘userId’ has been set to T/F’ should be printed in the console. The outcome should fit this equation ‘user.riskstatus = ((sum(user.contact(s).riskyWeight)) >= threshold)’	Same as expected
Test the ‘adjustRiskyWeight (userId)’ function from “server/index.js”	N/A	No error in each risk level, and message ‘The risk weight of user ‘userId’ has been adjust to ‘” should be printed in the console. The outcome should fit this equation ‘user.riskyweight = default * coefficient_riskStatus * coefficient_specialCase’	Same as expected
Test type: Integration testing Date: 2021-5-31		Responsible: Xuesong Ji Reviewer: Kaiyang Xue, Xilin Huang, Yumeng Zhang	