BCCH Test Plan

Team Info

Team Name: Future Star

Team Members: Nick Birtch, Lang Cheng, Jiayao Chen, Kaiti Mok Rong, Alexander Svatukhin, Jiawei Tan, Julie Zhu

Document Information

Revision History

Date	Version	Status	Prepared by	Comments
March 10th, 2020	0.0.1	Initial outline/notes.	Nick Birtch	N/A
March 11th, 2020	0.0.2	Drafting Functional/Non Functional Test Plans, Security Testing and Test Data Approach, Regression Testing.	Nick Birtch	N/A
March 11th, 2020	0.3	Finished Functional test plan	Jiayao Chen	N/A
March 11th, 2020	0.4	Finished Non-functional test plan	Jiayao Chen	N/A
March 11th, 2020	0.8	Working on Test Scripts, Security Testing and Test Data Approach, Regression Testing, Appendix	Jiayao Chen, Nick Birtch	N/A
March 12th, 2020	0.85	Working on Appendix	Jiayao Chen, Julie Zhu, Nick Birtch	N/A
March 12th, 2020	0.86	Working on Summary, Definitions etc.	Kaiti Mok Rong, Jiawei Tan	N/A
March 12th, 2020	0.87	Working on Summary	Lang Cheng	N/A
March 12th, 2020	0.90	Summary section finish - Nonfunctional, functional, Definitions etc.	Kaiti Mok Rong	N/A
March 12th, 2020	1.0	Finish Appendix, Complete	Nick Birtch	N/A
April 6th, 2020	1.1	Feedback Revision	Nick Birtch	Revised Summary

Document Control

Role	Name	E-mail	Telephone
Project Manager	Jiayao Chen	raymondyao1997@gmail.com	778-858-2077

Definition, Abbreviations & Acronyms

Black-box testing - Black-box testing is a method of testing where we test the functionality of the application without referencing the internal structure or working of the application

SAT - system acceptance tests

SQL - structured query language is a database management language for relational databases

UAT - user acceptance tests

White-box testing - White-box testing is method of testing where we test the functionality of the application while referencing the internal structure or working of the application

XSS attack - cross script attacks are a type of injection where malicious scripts are injected into the website

Summary

We will test the Functional, Security and Data components of our program using automated and manual testing and testing our Non-Functional components through manual testing. Automated and Manual testing will include both SAT and UAT using both the “black-box” and “white-box” test styles. These tests will help us validate if we have met the requirements outlined by BCCH and tests will help us verify each component of the program is functioning properly.

Tools

We will be using Selenium-Webdriver 4.0.0-alpha.5 (based on chrome version 80.0.3987.106 and chai ^4.2.0) to run the automated tests and Mocha 7.01 to run asynchronous tests. As for the data-related test, we use SQL 2.15.8.

Non-Functional

The Non-Functional components includes performance tests, assessments integrity tests, and mobile compatibility tests. The non-functional components will be using the manual testing approach through verification and validation.

(A) Pressure Testing (manual)

- Performance Test component
  - We will prepare 10k users uploading 20 times each, while making basic query every 10 seconds

(B) Database Performance (manual)

- Performance Test component
    - Upload a survey with at least two 10 minutes, 100MB sized videos in 1 minute

Functional, Data & Security

The Functional, Security and Data components will be utilizing both manual and automated testing. Various tests will be implemented to test these components of our system which are all listed in the table 1.1 below.

(A) Risk-Based Testing (manual & automated)

We will be doing our risk-based testing in our "*Security*" section where we will be testing our security features including, authentication bugs, query data anonymization, and common web application security tests. These features will be tested using the penetration testing approach where testers will attempt to attack the application by performing unusual browser operations or exploiting common security flaws.

Category	Components to be tested	Manual/Automated	Tool
Functional	Login & Logout Workflow	SAT: Manual & Automated UAT: Manual	Selenium & Mocha
	Upload an Assessment	SAT: Manual & Automated UAT: Manual	Selenium & Mocha
	Show Personal Data	SAT/UAT: Manual	N/A
	Manage Assessment Templates	SAT: Manual & Automated UAT: Manual	Selenium & Mocha
	Manage Survey Templates	SAT: Manual & Automated UAT: Manual	Selenium & Mocha
	Query User Data	SAT/UAT: Manual & Automated	Selenium & Mocha
	Export User Data	SAT/UAT: Manual	N/A
	Support Various File Types Upload	SAT/UAT: Manual	N/A
	Login Session Management Test	SAT/UAT: Manual	N/A
	Header Dropdown Menu	SAT/UAT: Manual	N/A

	Manage User	SAT/UAT: Manual	N/A
	Fill Out a Survey	SAT/UAT: Manual	N/A
Non-functional	Performance Test	SAT/UAT: Manual	N/A
	Assessment Integrity Test	SAT/UAT: Manual	N/A
	Mobile Compatibility Test	SAT/UAT: Manual	N/A
Security	Common Web Application Security Test	Manual & Automated	Mocha
	Data Privacy Test	Manual	N/A
Data	Data Test	Manual & Automated	SQL

Figure 1.1
Categorizing the tests and tools used to evaluate the program's functionality, security, nonfunctional and data components of the program.

Test Cycles

- Initial Test Cycle to ensure automated tests and manual testing processes working
    - Criterion:
        - 100% valid automated test responses (no errors or test stubs, ensure a pass/fail even if functionality not fully implemented)
        - Dry run and revision of manual testing processes to encapsulate all MVP functionality.
- Validation Test Cycle to ensure current system working as intended
    - Criterion:
        - 75% successful automated test responses (for current system functionality)
        - 75% successful manual tests (for current system functionality)
- Back End Test Cycle to ensure connection(s) successful
    - Criterion:
        - 85% successful automated test responses (all functionality)
        - 85% successful manual tests (all functionality)
- Final Test Cycle(s) before demo to ensure project stability
    - Criterion:
        - Ensure 100% of MVP functionality is tested by multiple parties
        - 95% successful automated test responses (all functionality)
        - 95% successful manual tests (all functionality)

Functional Test Plan

---

1. Login & Logout workflow

Testing Technique: Black box testing

Unit, SAT, UAT differences:

- SAT: a combination of automated testing and manual testing.
- UAT: manual testing.

Anticipated scripts:

- 2 happy-path testings:
    - login as a normal user
    - login as an admin user
- 3 negative-path testings:
    - wrong username or password
    - special characters input
    - logout and type URL to access protected pages

2. Upload an Assessment

Testing Technique: Black box testing

Unit, SAT, UAT differences:

- SAT: a combination of automated testing and manual testing.
- UAT: manual testing.

Anticipated scripts:

- 1 happy-path testing:
    - Upload an assessment with correct videos, pictures and surveys
- 2 negative-path testings:
    - wrong format files
    - upload an incomplete assessment
    - exit app during uploading.

3. Show Personal Data

Testing Technique: Black box testing

Unit, SAT, UAT differences:

- SAT: manual testing.
- UAT: manual testing.

Anticipated scripts:

- 1 happy-path testing:
    - correctly shows up user information
- 0 negative-path testing

4. Manage Assessment Templates

Testing Technique: Black & White box testing

Unit, SAT, UAT differences:

- SAT: a combination of automated testing and manual testing.
- UAT: manual testing.

Anticipated scripts:

- 4 happy-path testing:
    - see all existing assessments
    - archive an assessment
    - unarchive an assessment
    - add an assessment
- 1 negative-path testing:

    - toggle archive status in database, check results.

5. Manage Survey Templates

Testing Technique: Black & White box testing
Unit, SAT, UAT differences:

- SAT: a combination of automated testing and manual testing.
- UAT: manual testing.

Anticipated scripts:

- 2 happy-path testing:
    - see all existing surveys
    - add a survey
- 0 negative-path testing

6. Query User Data

Testing Technique: Black & White box testing
Unit, SAT, UAT differences:

- SAT: a combination of automated testing and manual testing.
- UAT: a combination of automated testing and manual testing.

Anticipated scripts:

- 3 happy-path testings:
    - build a filter query
    - build a group by query
    - send a plain text query
- 2 negative-path testings:
    - problematic input for query field
    - send a problematic query

7. Export User Data

Testing Technique: Black box testing
Unit, SAT, UAT differences:

- SAT: manual testing.
- UAT: manual testing.

Anticipated scripts:

- 1 happy-path testing:
    - export query result as a csv
- 1 negative-path testing:
    - export empty result set as a csv

8. Support Various File Types Upload

Testing Technique: Black box testing
Unit, SAT, UAT differences:

- SAT: manual testing.
- UAT: manual testing.

Anticipated scripts:

- 2 happy-path testings:
    - upload videos in format other than MP4

    - upload pictures in format other than JPEG
- 2 negative-path testings:
    - upload a non-video file should be impossible through UI
    - upload a non-picture file should be impossible through UI

9. Login Session Management Test

Testing Technique: White / Black box testing
Unit, SAT, UAT differences:

- SAT: manual testing.
- UAT: manual testing.

Anticipated scripts:

- 1 happy-path testings:
    - stay as logged in for a period of time
- 1 negative-path testing:
    - try to hit protected endpoints after login expires.

10. Header Dropdown Menu

Testing Technique: Black box testing
Unit, SAT, UAT differences:

- SAT: manual testing.
- UAT: manual testing.

Anticipated scripts:

- 2 happy-path testings:
    - can see header dropdown menu after logged in
    - cannot see header dropdown menu after logged out
- 0 negative-path testing:

11. Manage Users

Testing Technique: Black box testing
Unit, SAT, UAT differences:

- SAT: manual testing.
- UAT: manual testing.

Anticipated scripts:

- 2 happy-path testings:
    - be able to load new users from export
    - be able to remove a user with all its uploaded data
- 0 negative-path testing:

12. Fill Out a Survey

Testing Technique: Black box testing
Unit, SAT, UAT differences:

- SAT: manual testing.
- UAT: manual testing.

Anticipated scripts:

- 4 happy-path testings:
    - fill out a survey on desktop browsers

    - fill out a survey on phones
    - try all kinds of survey questions
    - re-enter saved survey with previous answers loaded
- 3 negative-path testing:
    - save incomplete/blank survey
    - invalid time input
    - leave multiple choice questions blank

Non-Functional Test Plan

---

1. Performance Test

Testing Technique: Black box testing.
Unit, SAT, UAT differences:

- SAT: a combination of automated testing and manual testing.
- UAT: manual testing.

Anticipated scripts:

- 2 happy-path testings:
    - prepare 10k users with 20 uploads each, do a basic query in 10 seconds.
    - upload a survey with at least two 10 minutes, 100MB sized videos in 1 minute
- 0 negative-path testing:

2. Assessment Integrity Test

Testing Technique: Black box testing.
Unit, SAT, UAT differences:

- SAT: manual testing.
- UAT: manual testing.

Anticipated scripts:

- 1 happy-path testings:
    - a successful upload should have all data in the database and blob store.
- 1 negative-path testing:
    - a failed upload (network failure, user interruption) results in partial data marked as archived in the database. Archived data will not be shown by default.

3. Mobile Compatibility Test

Testing Technique: Penetration testing.
Unit, SAT, UAT differences:

- SAT: manual testing.
- UAT: manual testing.

Anticipated scripts:

- 1 happy-path testings:
    - all UI is user friendly on desktop browsers.
    - all normal user UI is user friendly on iOS and Android
- 0 negative-path testing:

Test Scripts

Functional Scripts:

1. Login & Logout workflow
This will be a combination of automation tests written in selenium and manual testing. The automation tests will mimic users filling in the username and password input boxes and then clicking the login/logout button. After that it will try to find webpage elements that only exist after login as a proof of successful login. When it comes to manual testing, two testing normal user accounts and one admin user account were provided so that one can verify login status by trying to log into the application.

2. Upload an Assessment
This will be a combination of integration tests and manual testing. The integration tests will try to verify the corresponding data records are created in the database. This is done by first initiating an upload request to the application, and then requiring the corresponding data entries through queries. The manual testing, on the other hand, focuses on the persistence of media files in the system. It is done by making a real upload manually and then log into the system backend to check the existence of specific files.

3.Show Personal Data
This is a fairly straightforward feature and can be verified manually easily. After logging into the application as a normal user, one can click the "**info**" button on the dashboard to check whether the side sheet shows up and the content is correct.

4. Manage Assessment Templates
This test involves both black box and white box manual testing. In black box testing, one can log in as an admin user and then create/archive an assessment from scratch, after that it can log in as a normal user to see whether the corresponding assessment shows up/disappears. In white box testing, one can toggle the archived data field of AssessmentTemplate table and verify whether the assessment shows up / disappears on a normal user side.

5. Manage Survey Templates
This test involves both black box and white box manual testing. In black box testing, one can log in as an admin user and create a survey, after that it can create a new assessment requiring that survey. Then it can log in as a normal user and sees the new assessment shows up with the new survey attached. In white box testing, one can add/delete an entry in HasSurvey table to "link/unlink" a survey to an assessment. This should reflect on the normal user side.

6. Query User Data

This test will be a combination of integration tests and manual testing. The integration tests will first setup test data on the database and then send a query request to the application. Then they verify whether the response is correct. In manual testing, one can log in as an admin user and go to the query data section to make up a query and send it.

7. Export User Data

This is a fairly straightforward feature and can be verified manually easily. Log in as an admin user, send a query in the query section and click the "**export**"button. One should expect a csv file is downloaded and can open it to verify the result.

8. Support Various File Types Upload

This is a pure client side feature as this is in general hard to detect if users do it on deliberately. As a normal user, one can only select files with correct file extensions from the file explorer.

9. Login Session Management Test

By default, once logged in, a user does not have to log in again within a week. In manual testing, one can verify this by accessing the application after a week. The application also exposes login expiration time range as a configuration option so it can be verified more quickly.

10. Header Dropdown Menu

This is a pure client side feature and can be tested manually easily. Without logging into the app, there shouldn't be a dropdown menu on the header while the menu should be there once a user logs in.

11. Manage Users

The application allows admin users to import normal users from a csv file or permanently removes a user if she/he doesn't want to disclose her/his data any more. By removing a user all his/her associated data will be removed from the database. In manual testing, one can prepare a csv file with correct format and try to import the users into the application. In integration testing, the test will first setup test assessment data in the database, and then query some data and record down the number, after that it will send a "remove user" request to the application. In the end, it will rerun the same query and checks whether the data result is one less than the original.

12. Fill Out a Survey

By logging as a normal user, one can try to fill out a survey in an assessment. it should try to save a survey with wrong blank input, type in the wrong time format. Also, once a survey is successfully saved, it should try to reopen the survey to see whether previous results get restored.

Non-Functional Scripts:

1. Performance Test

This test involves a combination of automation scripts and manual tests which will be used to validate the system performance. The automation will handle resetting the database environment and creating the requisite test data.We will then manually test performance by monitoring our performance benchmarks while executing basic queries and uploading surveys, logged in as an admin and patient respectively.

2. Assessment Integrity Test

To ensure assessment data integrity, we will manually test uploading files and check the data stored in the database. Logged in as a Patient user, we will upload data normally and check the database and blob store for the appropriate records and files. We will also test attempting to fail uploads by disconnecting from the network while uploading and ensuring that any partial data is properly hidden. As a final check, we will login as an Admin user and execute queries to ensure that only the correct, expected data is shown.

3. Mobile Compatibility Test

As a high priority stretch goal, we will perform extensive manual tests of the UI on Android and iOS mobile devices. We will login as Admin and Patient users on several mobile devices (at least one Android, at least one iOS) and navigate through each page and interaction to ensure that all functionality is replicable on mobile. Ideally, while we gather test feedback and improve the mobile UI, we will validate that the experience is smooth, intuitive and user friendly.

Security Testing and Test Data Approach

Security Testing:

Our main security focus consists of two main aspects: **common web application security issues** and **data privacy issues** during querying data. Our main approach will be penetration testing, where testers will try to attack the application using common potential security flaws, or deliberately conduct unusual browser operations .

Common web application security testing:

- SQL Injection attack: for all text input boxes on a webpage, try to type in raw SQL query and send it. For some particular input fields (i.e. username & password), try to form a real attack SQL query and send it.
- XSS attack: for all text input boxes on a webpage, try to type in inline JavaScript code or HTML elements and send it.

- Man in the middle attack: this is in general hard to test. We enable HTTPS for our project and assume this is properly guarded by the SSL protocol.

Data privacy testing:

- Authentication bugs:
    - without logging as an admin user try to access admin user features by typing in URL directly.
    - login as an admin user, log out, and then use the "back" button on the browser to get back to the admin dashboard, try to access different features.
- Query data anonymization:
    - In the UI query section, check name is not an interactable column.
    - When name anonymization feature is enabled, verify that all users stored in the database will have a nickname instead.

Test Data Approach:

As detailed in our non-functional test plan for performance, we will create 10,000 patient records with 20 activity records per patient. We will also add 2-3 100MB video files per patient record. To create these records, we will use a sql script that inserts dummy information while properly incrementing and associating the correct IDs to records in connected tables. Once the database has been successfully populated, we can create backups that can be used to reset the database to the initial populated state for ease of use.

Regression Testing

Regression Testing serves to ensure stability on existing functionality when new changes are introduced. The core features we have identified as suitable for a regression test pack are:

- Login Functionality and Authentication
- Upload Assessment
- Manage Assessment Templates
- Query User Data
- Export User Data

We have selected these features as they are high priority, MVP functionality that has been requested by BCCH and are integral to any database management system. This application would be fundamentally different and inoperative without these functionalities and all incorporate automated testing which suits a regression testing pack.

Appendix

ALL Details of test script approach
Manual Tests: Test conditions, expected output should be outlined
Automated Tests: Input/Output and what method you will use to automate should be described

<u>Test Scripts</u>
Functional Scripts:
1. Login & Logout workflow
Automation:

1) run test/test.js

Expected Output:
A browser should pop up mimicking user input and navigating. No errors should be thrown.

Manual testing:

1) Enter valid Normal user's username & password
2) Click "login" button
3) User is logged in successfully and is directed to user dashboard
4) Click "logout" button
5) User is returned to login page

1) Enter valid Admin user's username & password
2) Click "login" button
3) User is logged in successfully and is directed to admin dashboard
4) Click "logout" button
5) User is returned to login page

2. Upload an Assessment

1) Login as Normal user
2) Click "upload" button on the dashboard
3) User is directed to Upload page
4) Upload file data and complete multiple assessments
5) **Expected Output: Check the database for the assessment records attached to the Patient user IDs**
6) **Expected Output: Check the blob store for the files uploaded in assessments**
7) Navigate to Upload page

8) While uploading an assessment, attempt the following:
    a) Upload files of wrong format
    b) Uploading incomplete assessments
    c) Exit application during upload.
9) **Expected Output: Check the database and blob store and ensure no erroneous data has been uploaded.**

3.Show Personal Data

1) Login as Normal user
2) Click "Info" button on the dashboard
3) The right side sheet should show up with basic personal information.
4) Clicking the "X" button or area outside of the sheet should result in closing the side sheet

4. Manage Assessment Templates

1) Login as Admin user
2) Navigate to Manage Assessment Templates
3) **Expected Output: See all existing assessments (cross reference with database)**
4) Add three new Assessments with visibility for a Patient user
    a) 2 Archived Assessments
    b) Non-Archived Assessment (Normal)
5) **Expected Output: Check the database for relevant Assessment records and Archived status.**
6) Login as Patient user
7) Navigate to Upload data
8) **Expected Output: User able to fill out non-archived Assessment**
9) **Expected Output: User unable to see or fill out archived Assessments**
10) Login as Admin user
11) Navigate to Manage Assessment Templates
12) Unarchive one of the archived Assessments through UI
13) Toggle archive status in database for other archived Assessment
14) Login as Patient user
15) Navigate to Upload data
16) **Expected Output: User able to see and fill out all three new Assessments**

5. Manage Survey Templates

17) Login as Admin user
18) Navigate to Manage Survey Templates
19) **Expected Output: See all existing surveys (cross reference with database)**
20) Add a new Survey with visibility for a Patient user
21) **Expected Output: Check the database for relevant Survey and Question records.**

22) Login as Patient user
23) Navigate to Upload data
24) **Expected Output: User able to fill out new Survey in an assessment.**

6. Query User Data

Automation:

1) Setup database by sourcing sql file query_test_data
2) Run test/integration/QueryTest.ts

    Expected output:
    Sending query "SELECT * FROM Video WHERE assess_id = 1"
    complete comparison.
    Sending query "SELECT Count(*) FROM SurveyAnswer JOIN Survey ON SurveyAnswer.survey_id = Survey.id GROUP BY answer"
    complete comparison
    all tests passed

Manual testing:

1) Login as Admin user
2) go to Query Data section
3) send a filter query, verify the result
4) send a group by query, verify the result
5) send an arbitrary query, verify the result

7. Export User Data

1) Login as Admin user
2) Go to Query Data section
3) Execute a query to get some data.
4) Click the "Export" button.
5) Check a csv file is downloaded by the browser.

8. Support Various File Types Upload

1) Login as Normal user
2) Go to Upload section
3) Select an assessment that requires Video and Picture
4) Click the "Upload" button in Video section
5) The file explorer that pops up should allow selection of all video types. Other types of files should be hidden
6) Click the "Upload" button in Picture section
7) The file explorer that pops up should allow selection of all picture types. Other types of files should be hidden.

9. Login Session Management Test

1) In code or setting file, set session expire time range to 2 minutes.
2) Log in as Normal user.
3) Refreshing the browser should stay on the same page.
4) After 2 minutes, refresh the browser again.
5) The user should be directed back to the login page.

10. Header Dropdown Menu

1) On the login page, there should be no header dropdown menu.
2) Login as Normal user
3) There should be a header dropdown menu. Clicking the icon will show up "log out".
4) Logout. The dropdown menu should disappear.

11. Manage Users

1) Login as Admin user
2) Navigate to Manage Users
3) Create dummy Patient user data
4) Load new users (**Expected Output: Check the database for the Patient records added)**
5) Login as added Patient user
6) Navigate to Upload page
7) Upload multiple surveys
8) Login as Admin user
9) Navigate to Manage Users
10) Select added Patient user to remove (**Expected Output: Validate the database does not contain removed Patient records and associated data)**

12. Fill Out a Survey

1) Login as Patient user
2) Navigate to Upload page
3) Upload multiple surveys normally with various question types
4) **Expected Output: Check the database for the survey records attached to the Patient user IDs**
5) Navigate to Upload page
6) While uploading a survey, attempt the following:
    a) Saving an incomplete survey
    b) Using an invalid time input
    c) Leaving multiple choice and required questions blank
7) **Expected Output: Check the database and ensure no erroneous data has been uploaded.**

Non-Functional Scripts:

1. Performance Test

Automation:

We will use a sql script that inserts dummy information while properly incrementing and associating the correct IDs to records in connected tables. Once the database has been successfully populated, we can create backups that can be used to reset the database to the initial populated state for ease of use.

The automation above will handle resetting the database environment and creating the 10,000 patient records with 20 activity records per patient. We will then manually test performance by executing basic queries in ~10 seconds and uploading surveys with at least two ~10 minute, 100MB sized videos in 1 minute while logged in as an admin and patient respectively.

Test Condition: Run database environment reset automation script

5) Login as Admin user
6) Navigate to Query page
7) Execute several basic queries **(Expected Output: Query responds in less than ~10 seconds)**
8) Logout
9) Login as Patient user
10) Navigate to Upload page
11) Upload multiple surveys with at least two ~10 minute, 100MB sized videos **(Expected Output: Upload finishes in less than ~1 min)**

2. Assessment Integrity Test

Test Condition: N/A

8) Login as Patient user
9) Navigate to Upload page
10) Upload multiple assessments normally
11) **Expected Output: Check the database for the assessments records attached to the Patient user IDs**
12) **Expected Output: Check the blob store for the files uploaded in assessments**
13) Navigate to Upload page
14) Upload multiple assessments, while doing the following:
    a) Disconnect your device from the internet
    b) Close browser
    c) Upload corrupted data
15) **Expected Output: Check the database for the assessments records attached to the Patient user IDs, marked as archived.**
16) **Expected Output: Check the blob store for the files uploaded in assessments, marked as archived.**

17) Login as Admin user
18) Execute several basic queries on the data uploaded **(Expected Output: Only non-archived data should be shown)**

3. Mobile Compatibility Test

Test Condition: Usage of a Android or iOS mobile device or equivalent emulator

1) Login as Admin user
2) Navigate through each accessible page **(Expected Output: User Friendly UI)**
    a) Click all buttons, try flipping phone horizontally/vertically, try zoom/scrolling/swiping actions. **(Expected Output: UI is adaptive, stable)**
    b) Iterate through all admin functionality per page **(Expected Output: All functionality replicable and working as intended)**
3) Login as Patient user
4) Navigate through each accessible page **(Expected Output: Adaptive, User Friendly UI)**
    a) Click all buttons, try flipping phone horizontally/vertically, try zoom/scrolling/swiping actions. **(Expected Output: UI is adaptive, stable)**
    b) Iterate through all user functionality per page **(Expected Output: All functionality replicable and working as intended)**
    c) Extensive focus on filling out surveys and uploading data as the main focal point of mobile compatibility **(Expected Output: All functionality replicable and working as intended, easy to use)**