

Tan D. Duong
COSC 412-102
Breathalyser Test Web App - Test Procedure

Content

Use Cases and Requirements Updated/Changes

Use Cases and Requirements from March 08, 2020

Use Cases and Requirements Updated (From April 1, 2020)

Reasons for Changes of Requirements

Unit Tests

For Web App

For Users

End - to - End System Test Procedure

Requirement Review Test

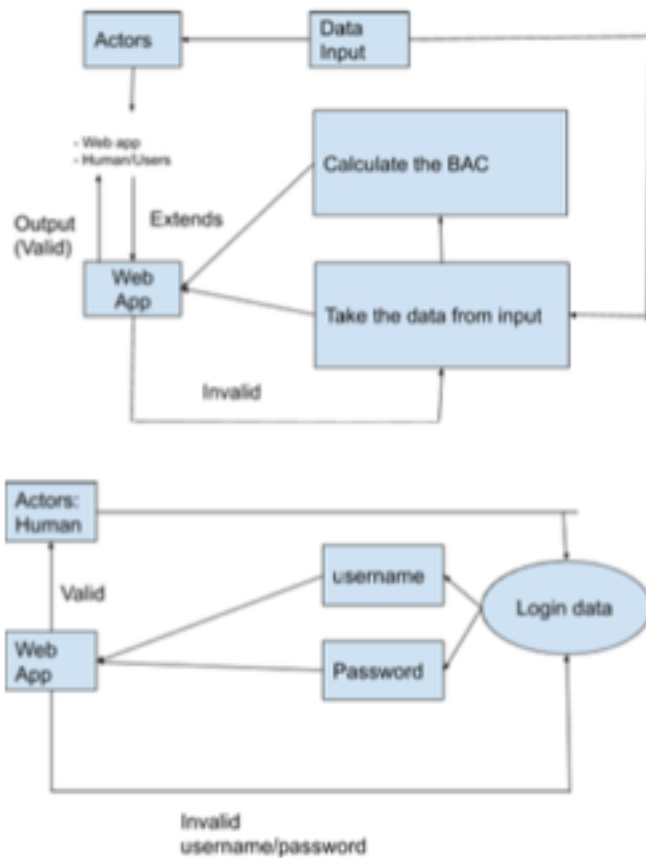
Input Tests

Loads

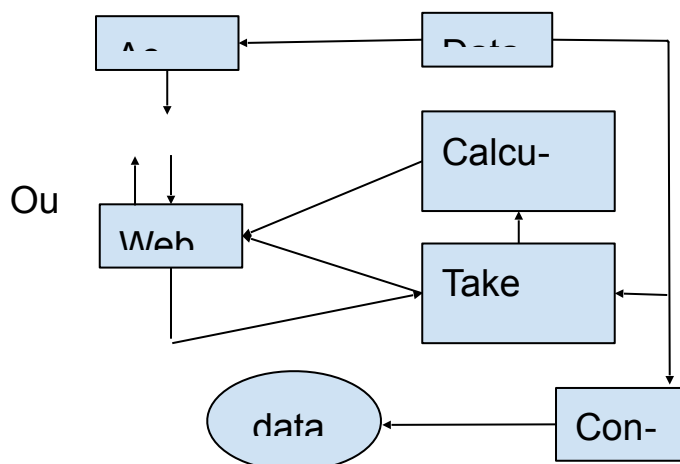
Testing the App Step-by-Step

Use Cases and Requirements Updated/Changed

Use Cases and Requirements from March 08, 2020



Use Cases and Requirements Updated (From April 1, 2020)



Reasons for Changes of Requirements

- **Lack of Backend experience:** The Developer does not have high-quality skill in building the Backend system and requires time for researching. Due to the deadline issue, this requirement was temporarily set aside, it will be updated in the future.
- **Simplify the App:** This Web App was built as a tool to help people, so the most important part is to guarantee the algorithm of calculating running well and fast, and keep it in a single page. The login page somehow is not much necessary, but it does not completely be gotten rid of.
- **Time Limit:** Finishing the app before the due date is the first priority. So the developer needs to consider which features need to be implemented in the web-page.
- **Temporary Solution:** The Solution to replace the login page is to store the database from the users by connecting it with Firebase, through the contact form. This solution is faster and less code, but it contains data limit issues. The Developer listed it as the temporary strategy before finding better ways to solve it.

Unit Tests

For the Web App:

- Guarantees all the code is clean and works. All the bugs need to be fixed immediately if found. Files should return well on the screen with the features from CSS and JS.
- Guarantees the tools run well and return fast, accuracy results. Invalid data input, if found, should be popped up and announced to the users in the fast way.
- Guarantees the app be responsive on any devices. Use Develop Tools for this test, CSS skills play an important role in this test since it was built from scratch.
- Guarantees host server works well while deploy on the Internet

For Users:

- Guarantees the Measure and Contact form can run and return results.
- Guarantees the messages from the users be sent.

- Guarantees all the links should be clicked and visited.

End - to - End System Test Procedure

Requirement Review Test (workflow)

First, users search for the URL of the app: <https://testbac.com>. When the app shows up, the navbar will show all the sections of the website: **Home, About, Service, Get BAC, Articles, Author, and Contact**. **Home** is the top of the page, **About** and **Service** provided some information about the app and statistics related to drunk driving. **Get BAC** is the section for Users to input the data and receive the result. **Articles** show some related studies and articles. The **Contact** form lays in the bottom of the page, where Users type their name, email and message/feedback. This website is single page.

Input Test

Typed Input/Clicked method (**Get BAC** and **Contact**)

while(GetBAC)

 if(!blank) return results;

 Else send out error message;

 If (reset) return new blank;

End while

while(Contact)

 if(AllBlank) return nothing;

 Else if(!AllBlank)

 if(Submit.clicked())

 return error message;

 redo();

 Else return nothing;

End while

Usability

The application will be built for users as adults and elders, who are eligible for drinking liquor, so the usability should be easy to follow and the user can access all of the resources of the app/website. For more information, users can send feedback to the admin/developers to fix or improve the app performance if needed.

Loads

Developer expects this website will not contain any lags or bugs while displaying on the users' screen. Real-life loads of the site is the first priority.

Test Procedure: Breathalyser Web App

“**Checked**” means to pass the test.

Step 1 - Checking the load when open the website

Requirements: fast, no lags, shows navbar, body and footer (Checked)

Step 2 - Checking the UI

Requirements: Nice visual, friendliness, fonts easy to read (Checked)

Step 3 - Checking the Responsiveness

Requirements: Nice performance guarantees all the features fit well in any screens and devices (Checked)

Step 4 - Checking the Links of Section

Requirements: When click a link on the navbar, the web should jump right into that section (Checked)

Step 5 - Checking any links on the body

Requirements: no dead links, links should suitable with the title or icon displayed (Checked)

Step 6 - Testing the **Get BAC**

Requirements: Data should fit the type of question, automatically send the results if valid data submitted, or return error alert if invalid (Checked)

Step 7 - Testing the **Contact**

Requirements: Data should fit the type of question, automatically send the results if valid data submitted, and also reset the form. Data should be sent and stored into the Firebase project (Checked)