# Software Design Specification for Quality Assessor Tool

Authors: Marwa Hachem, Iman Saleh, Farzana Fariha, Connor McDonough, Samara Sarmiento

CIS 375 | Fall 2019

## Revision Chart

| Version | Primary Author(s) | Description of Version | Date Completed |
|---|---|---|---|
| 1 | Marwa Hachem, Iman Saleh, Farzana Fariha, Connor McDonough, Samara Sarmiento | Document draft prepared before peer reviews. | 11/14/19 |
| 2 | Marwa Hachem, Iman Saleh, Farzana Fariha, Connor McDonough, Samara Sarmiento | Document draft after peer reviews. | 11/17/2019 |
| 3 | Marwa Hachem, Iman Saleh, Farzana Fariha, Connor McDonough, Samara Sarmiento | Final Document Revisions | 12/16/2019 |

# SOFTWARE DESIGN SPECIFICATION for QUALITY ASSESSOR

Marwa Hachem, Iman Saleh, Farzana Fariha, Connor McDonough, Samara Sarmiento

## 1.0 Introduction

This section provides an overview of the entire design document. This document describes all data, architectural, interface and component-level design for the Quality Assessor Tool.

### 1.1 Goals and objectives

Overall goals and software objectives are described for the Quality Assessor Tool.

The goal for creating the Quality Assessor Tool is for a user to test their website in terms of its quality. In other words, the user will be able to gauge what aspects of their website are harmful and what is helpful to the average visitor. The assessor will be in a question and answer format with the aim that any user can understand and answer each question confidently. In order for this to happen, widely understood diction will be used, and sentence formatting will need to be carefully chosen. The tool will ask questions based on: content, function, structure, usability, navigability, performance, compatibility, interoperability, and security (Pressman, Maxim).

### 1.2 Statement of scope

A description of the software is presented. Major inputs, processing functionality, and outputs are described without regard to implementation detail.

#### 1.2.1 User Stories / Use Cases

**Use Case 1:** User can enter the site
**Primary Actor:** Website Visitor / User
**Preconditions:** A user must have an internet connection
**Description:** As a website visitor/user I want to be able to view the Home page of the Quality Assessor Tool that describes the purpose of the website. Upon landing on the website, I want to see a navigation page with all the web pages on the top of the page.
**Acceptance Criteria:** The Home page has been loaded up, the website description is displayed.

**Use Case 2:** User can start the questionnaire
**Primary Actor:** Website Visitor / User

**Preconditions:** A user must have internet connection, and either selected the assessment tab or clicked begin assessment on the home page.
**Description:** As a website visitor/user I want to be able to begin the Quality Assessment. I want to begin the first page of questions to begin assessing my website.
**Acceptance Criteria:** The assessment has begun, the first page of the questionnaire has loaded up.

**Use Case 3:** Going through the questionnaire
**Primary Actor:** Website Visitor / User
**Preconditions:** A user have have an internet connection, and have filled out the information to begin the questionnaire
**Description:** As the visitor/user begin the questionnaire they will be given questions to answer (in forms of a scale from 1-5). Each section will have 3-4 questions with a total of 9 sections. The user will be able to go back to a section using the back button and will be able to move on to the next section with the next button.
**Acceptance Criteria:** Answer the user enters are stored and accurate.

**Use Case 4**: Confirmation Page
**Primary Actor**: Website Visitor / User
**Preconditions**: The assessment must have been completed to reach this stage.
**Description**: The confirmation page will consist of a score on the radio button.
**Acceptance Criteria**: A selected radio button is displayed for each question in all sections and are editable by the user.

**Use Case 5:** Results Page
**Primary Actor:** Website Visitor / User
**Preconditions:** The assessment must have been completed to reach this stage.
**Description:** The confirmation page will consist of a numeric score based on the answers of the questionnaire.
**Acceptance Criteria:** A score for each section is displayed as well as an overall score.

**Use Case 6:** Navigation through the site
**Primary Actor:** Website Visitor / User
**Preconditions:** The user must have access to each page on the site and navigate through them with a navigation bar.
**Description:** As the visitor/user enters the site, they must be able to move through the pages from the navigation bar. Each link will redirect to appropriately named page.
**Acceptance Criteria:** The proper page has come up after a user clicks on it in the navigation bar.

#### 1.2.1.1 Requirements

The following requirements are the main features needed for the Quality Assessor Tool. The team is aware of the constraints (section 5.0), and thus has limited these necessary functionalities for the website.

1. Testing the quality of a website through Q&A format
2. Tests through the following sections: content, function, structure, usability, navigability, performance, compatibility, interoperability, and security
3. Allow users to answer the questions with a scale of 1 - 5. Where features with a 1 are non existent, and features with a 5 are available and useful.
4. Allow users to select Not Applicable or N/A as an answer for each question
5. Allow users to see the remaining sections left to complete
6. Allow users to return to a previous section
7. Users must answer all questions in order to print or get final results.
8. Allow users to edit their selections at the end of the survey
9. Allow users to print their results
10. Allow users to restart the survey
11. Display a summary of the user's answers
12. The users will receive feedback on the quality of their website
    a. Low Quality (low scores on categories)
    b. Medium Quality (average scores from each category)
    c. High Quality (high scores)

#### 1.2.1.2 Additional Requirements (Optional)

The following requirements are meant for future updates of the website that may not be implemented upon creation of the Quality Assessor Tool. These features are quality additions to the website but are not necessary for the first version of the tool.

1. Allow users to give feedback on the website assessor tool

2. Website Assessor tool should be available on all web platforms

#### 1.2.3 Inputs

1. The user will select a choice between 1-5 per question
2. The user can select “N/A” as a choice
3. The user can click to go back
4. The user can click to go forward
5. The user can click to print the results
6. The user can click to restart the assessment
7. The user can click the navigation bar to go to
    a. Home, Assessment, Help pages

#### 1.2.4 Outputs

1. Displays the user’s answers (confirmation)
2. Display the user’s results
3. Display a numeric score result
4. Display feedback on the user’s website
    a. In the form of a grading from low quality to high
5. Display the page the user clicked on the navigation bar
6. Displays website home page

### 1.3 Software context

The software is placed in a business or product line context. Strategic issues relevant to context are discussed. The intent is for the reader to understand the 'big picture'.

#### 1.3.1 Target Audience

The targeted audience for the Quality Assessor Tool is all stakeholders involved in a website. In other words, the website should appeal to investors, programmers, and even visitors of a given website. Acknowledging that this is a vast audience we aim to gear towards, the questionnaire will need to have diction that any member of the audience can understand and answer confidently. So, the user should be able to have the website at hand and judge it against the questionnaire. Another point is that the questionnaire should subscribe to the main components that each website should have. In the unique cases where the question doesn't apply to a given website, the user has the option of Not Applicable (which will not impact their score).

### 1.3.2 Big Picture

The overarching concept of the Quality Assessor Tool is to help users improve and judge a website based on the feedback that the tool provides. The tool itself is the questionnaire that the user will complete to evaluate a website. Once the calculations based on each question are complied, the feedback will be given to the user. The feedback will be useful to the user as it will display a score of the website, as well as the points to improve on. From there, the possibilities are endless and what the user chooses to do with the feedback is completely up to them. The tool is a very powerful one for a given user: someone who will take the feedback and choose to improve a website because of it. Even if this scenario is not the case, the tool will give useful feedback to any interested party. Overall, the Quality Assessor Tool is bound to be accessed and, from there, generate feedback that benefits any of the targeted audience and more.

## 1.4 Major constraints

Any business or product line constraints that will impact the manner in which the Quality Assessor Tool is to be specified, designed, implemented or tested are noted here.

First and foremost, the Quality Assessor Tool faces many constraints that will impact every aspect of the product. The tool will be designed and implemented by a team of 5 members. From this, there is bound to be disagreements and at least one disappointed party with any given result that the project requires. Especially with subjective aspects such as design, a divergence in opinion is expected. In turn, this will impact the design and implementation of the tool which may negatively reflect on the product as a whole (making it a product line constraint).

Another major constraint that we face that impacts the overall product is time. With the given amount of time from the stakeholder, the team has many integral aspects of the product that are expected to be complete and working at the end of the period. The entire working product is expected on December 15, which gives the team roughly 4 weeks to compile a test plan and the product itself (after this phase, the SDS, is due). To add to the tight time constraint, the team is inexperienced with web development. Of course, this will only make the time frame even tighter because everyone will need to learn the languages that come with creating websites and implement them effectively. Overall, time is a very impactful constraint that the team is aware of and will surely impact the product itself.

## 2.0 Data design

A description of all data structures including internal, global, and temporary data structures.

For creating  the Quality Assessor Tool, we will be using Javascript, HTML, and CSS programming languages. Despite the team's inexperience with these languages, we will aim to improve our knowledge of them throughout the process. Please keep that in mind for the following section.

### 2.1 Internal software data structure

Data structures that are passed among components of the software are described.

**Array assessmentScores**

After every new section, the information from the temporarySection will be added to the assessmentScores, in the appropriate location to be calculated later.

**Integer assessmentResults**

After the assessment, the scores from the assessmentScores will be calculated and displayed as assessmentResults.

### 2.2 Global data structure

Data structures that are available to major portions of the architecture are described.

For the tool, there are no set-in-stone global data structures that are necessary to build it. However, upon research, websites have a tree structure within the HTML files called the DOM (Document Object Model) that allows Javascript to access all of its elements (w3schools). The team views this as a powerful aspect of the language to attempt to implement for our website. In that case, the Javascript HTML DOM serves as the only global data structure we will use at this point.

A global data structure that is more obvious in our project is the global.**sessionStorage** functionality in JavaScript. From this, we are able to store the user's choices in their temporary memory using a key and value. This is basically using cookies to remember the choices the user makes later on.

### 2.3 Temporary data structure

Files created for interim use are described.

Not Applicable.

**Array temporarySection**

On every new section, whenever the user answers a question it will be added to temporarySection. After that section, those results are added to the assessmentScores array. This data structure will be available to assessment pages only. Every time a brand new section is loaded, it will clear out the temporarySection and start empty.

### 2.4 Database description

Database(s) created as part of the application is(are) described. Since the Software Quality Tool doesn't require data to be saved for later use, there is **no need for a database**. However, in future versions of the tool, the team wishes to implement a feedback option, for which we will discuss integration of a database when that time comes. As for now, the prototype and the initial release of the tool will not have any databases.

### 2.5 Data Dictionary : JavaScript

| Name | Data Type | Location | Description |
|---|---|---|---|
| sectionNum | var(number) | script.js function: handleNext, parameter | This parameter will tell us which section called the function handleNext. |
| qName | var (string) | script.js function: handleNext | When the next button is clicked, this function is called. The variable will store the current question's name temporarily. |
| value | var (number) | script.js function: handleNext | From the current question, the user chooses a radio button that has an attached number value. This variable holds the number of the chosen answer. |
| A1Path | var(string) | script.js function: handleNext | This variable holds the path from which the function was called. |
| temp | var(string) | script.js function: handleNext | This temporary string holds the section number from which this function was called. |
| x | var(number) | script.js function: handleNext, in a for loop | This variable is responsible for looping through the answers that the user chooses on a given page. Will loop 3 times because each section has 3 questions. |
| radioBtns[] | var(nodelist) | script.js function: handleNext, in a for | This variable is a node list that holds all the radio button elements for a |

| | | | |
|---|---|---|---|
| | | loop | given question. There will always be 6 radio buttons, and this will hold their nodes. Used to see which node was chosen by the user. |
| i | var(number) | script.js function: handleNext, in a for loop within the initial loop | This variable is responsible for looping through each radio button option for a given answer. There are always 6 radio buttons, so this will reach the number 5. |
| qName | var(string) | script.js function: loadRadioBtn | Much like previous, will hold the identifier of the given question. |
| A1Path | var(string) | script.js function: loadRadioBtn | Holds the path name of the given section. |
| temp | var(string) | script.js function: loadRadioBtn | Holds the name of the given section to compare to path name. |
| x | var(number) | script.js function: loadRadioBtn | Responsible for looping through all the answers on a given section. 3 questions per page means 3 loops. |
| radioBtnID | var(string) | script.js function: loadRadioBtn | This variable gets the radio button's ID name to be used within the loadRadioBtn function. |
| radioBtns | object | script.js function: handleNext, | This variable gets the radio button's specific name to be used within the handleNext function. |
| resultsPath | var(string) | Global variable | This variable is used to match what page the script is on, using the path. |
| totalScore | var(number) | Results If Statement | This variable is what the total score of the questionnaire, which can be used to determine the quality of the website. |
| s | var(number) | Results If Statement | This variable is responsible for looping through each section. It will loop 9 times due to having 9 sections |
| resultMessage | var(string) | Results If Statement | This is a string that will be outputted on the results page which will tell |

| | | | |
|---|---|---|---|
| | | | the user the quality of their message. This message is determined by the range of the totalScore. |
| qName | var(string) | script.js function: loadForConfirmation | This variable will hold the identifier of the given question. |
| A1Path | var(string) | script.js function: loadForConfirmation | This variable holds the path name of the given section. |

**3.0 Architectural and component-level design**

A description of the program architecture is presented.

**3.1 Program Structure**

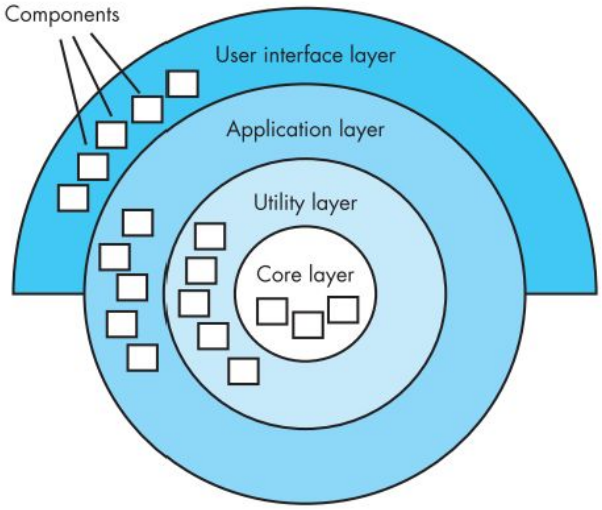
A detailed description of the program structure chosen for the application is presented.

Layered Architectural style: is one of the most common architectural styles. Functions are organized into horizontal layers, these layers execute particular roles inside of the application. The layered architectural has a number of different layers that are defined. So the developers could have as many layers as they need while developing the application. Each layer is independent from the others, which provides isolation from each other and a hierarchy. Once an action is invoked it cannot be changed, the action would need to be changed and invoked again.

When applying the layered architectural style to our Website Quality Assessor tool, the layers will be the following: User Interface layer, Application layer, Utility layer, and Core layer. The first layer of User Interface has components which would be the webpages on the website. This layer handles all user interface and browser communication logic, The User Interface layer does not need to know how to get user data. The User Interface layer in our model only needs to display the information on a screen in a particular format. The next layer is the Application layer where the components are the radio buttons and input buttons for the questionnaire. This layer does not need to worry about how to format customer data. Application layer worries about getting the data from user inputs  and passes the values. The Utility layer collects the data and prepares it to apply the logic to calculate the results to the Core layer. The calculations are created and now have to be stored and prepared to be sent back up through the layers. The Core layer stores the data of the saved inputs for the user, so it can be loaded up when needed.

### 3.1.1 Architecture diagram

A pictorial representation of the Layered Architectural Style is presented.



### 3.1.2 Alternatives

A discussion of other architectural styles considered is presented. Reasons for the selection of the style presented in Section 3.1.1 are provided.

The reason the team chose the Layered Architectural Style is because of its familiarity with software engineers. The flexibility of implementation and the use of layers which helps us organize the project. The Layered model makes the most sense when talking about a UI project. The outer layer would be components that service the user operations. The most inner layer, the core, would handle operating system interfacing. The layers in between are utility services and the software functions of the application. This model will be easy to implement within our software plan to create the quality assessor tool.

An architectural style the team considered, but didn't choose was the Call and Return Style. The primary reason we didn't choose this model is the drawback of the lack of ability to flow between the components in this model. Our system will need to be able to connect to each component in order to have the functionality of going back and forth in the questionnaire. With the call and return style, it would mean that we would lose the option to have a back a forward button, in our case, it would most likely result in the loss of data. This is a necessary requirement for a good user experience and a drawback that we couldn't look past.

Another architectural style we decided not to use was the Data Flow Architecture. The way we looked at it, the questionnaire doesn't work with much data for us to send through pipes and filters. The software will compute very simple math that will amount to very little data. For this architecture to work, data must be the biggest focus of the product. This is not the case for the Quality Assessor Tool. Our biggest focus will be the interaction between the UI and the application. The data processed in the layers will give the user the feedback they are looking for. That's why we went with layered rather than Data Flow.

## 3.2 Description for Component n (Home Page)

A detailed description of each software component contained within the architecture is presented. Section 3.2 is repeated for each of the n components.

### 3.2.1 Processing narrative (PSPEC) for component n

The user will open the assessment. The first page the user will see is the Home Page. The home page consists of a welcome screen and a description of the assessment the user is about to take. When the user is ready to begin the assessment questionnaire, they can click the start button.

### 3.2.2 Component n interface description.

Input: The only user input available on the home page screen is the start button and the tab at the top to navigate to another section.

Output: The output depends on the user's input on the home screen. The only two output options are: either displaying the actual assessment or launching the user's selected page.

### 3.2.3 Sub-Component n.m processing detail

A detailed algorithmic description for each sub-component within the component n is presented. Section 3.2.3 is repeated for each of the m sub-components of component n.

#### 3.2.3.1 Interface description

**Home Page**:

When the webpage is loaded, the user is prompted with a description of the assessor tiny tool. The navbar at the top of the page lists the home button, questionnaire button, and the help button. After the description of the tiny tool lays a start button that starts the questionnaire. In the lower right is the logo of the tiny tool.

#### 3.2.3.2 Algorithmic model (e.g., PDL)

*The pseudocode listing for sub-component m is presented.*

```
<!-- create the HTML/CSS doc -- >
<!-- print description of assessment  -->
<!-- create the start button under the description-->
<!-- add a link to the button to link to the questionnaire page -->
<!-- create the navigation bar on top -->
<!-- add page links to the navbar -->
<!-- insert the business logo (jpg image) at the bottom -->
```

### 3.2.3.3 Restrictions/limitations

The external environment and/or infrastructure that must exist for sub-component m to operate correctly is provided.

There aren't any external environment or infrastructure that will affect m to operate correctly.

### 3.2.3.4 Local data structures

The data structures used within sub-component m are presented.

There are not any data structures for the Home Page.

### 3.2.3.5 Performance issues

Information on topics that may affect the run-time performance, security, or computational accuracy of this sub-component are presented.

For the Quality Assessor Tool Home Page, the page consists of buttons and a navigation bar. This will cause run-time to be slower because of the many buttons and links, and we need the website to be functional. Another performance issue is that if this first web page doesn't load (low internet connection, etc.), then the entire assessment won't load. This is because the only way to access the assessment is to first get through the Home Page.

### 3.2.3.6 Design constraints

Attributes of the overall software design (including data structures, OS features, I/O, and interoperable systems) that constrain the design of this sub-component are presented.

The website assessment will also be used on all types of mobile devices, as long as they have an internet browser. For example, if a Mac user wants to access our website, they will use Safari.

The I/O features include the user clicking the start button, or selecting an option from the navigation bar at the top.

The interoperable systems is compatible with other OS and products and devices.

## 3.2 Description for Component n (Assessment Page)

A detailed description of each software component contained within the architecture is presented. Section 3.2 is repeated for each of the n components.

### 3.2.1 Processing narrative (PSPEC) for component n

This is where the user will begin the assessment. The user will be asked a series of questions per section, answering from a range from 1-5 from disagree to agree. The user will be complete the assessment once all the sections have been answered. The user can also select to start over, and move back or forward.

### 3.2.2 Component n interface description.

Input: The user will be selecting from choices 1-5 about an evaluation of their website. This is to help determine the quality of the website. The ranks will be based on questions about website performance, functionality, content, etc.

Output: The questionnaire will output each page of the form to the users. It will also output website results to users. In addition, an output includes when the user selects to print, it will print. If the user wants to go back, it can go backwards.

### 3.2.3 Sub-Component n.m processing detail

A detailed algorithmic description for each sub-component within the component n is presented. Section 3.2.3 is repeated for each of the m sub-components of component n.

#### 3.2.3.1 Interface description

**Questionnaire Page**:

The questionnaire page will begin to display a textbox with a description of what is included in the questionnaire. Then they will be able to invoke the begin button in the lower right corner of those inputs. Once the user inputs all necessary information they will begin the questionnaire. Each section will have three questions for the user to answer. These questions will be inputted with a scale from 1 (being disagreeing) to 5 (being agreeing).And the user can choose "N/A" if 1-5 does not apply. The user will be able to go back to the previous section or to the next section using the corresponding button. In the lower middle of the screen will displace the process bar of the questionnaire. At the end, the website will output a confirmation of the user's answers. After that, the results will be displayed and the user has the option to redo the test or print the results.

#### 3.2.3.2 Algorithmic model (e.g., PDL)

*The pseudocode listing for sub-component m is presented.*

```
Inputs:
// X depending on how many questions we decide per section
For i = 0 to X
        Set temporarySection [i] = answer;
End For


// Y depending on how many questions for the questionnaire
```

```
For j = 0 to Y
        Set assessmentScores [j] = temporarySection [i] ;
End For
```

Outputs: ResultsPage();

### 3.2.3.3 Restrictions/limitations

The external environment and/or infrastructure that must exist for sub-component m to operate correctly is provided.

There aren't any external environment or infrastructure that will affect m to operate correctly.

### 3.2.3.4 Local data structures

The data structures used within sub-component m are presented.

Questionnaire Page:

Array temporarySection

Array assessmentScores

### 3.2.3.5 Performance issues

Information on topics that may affect the run-time performance, security, or computational accuracy of this sub-component are presented.

For the Quality Assessor Tool assessment questions, the information within the internal software data structure is temporarily stored in one array and it is also then stored into another permanent array. Furthermore, to view the information for one array requires one nested loop, but with two requires double the amount of time. This isn't efficient as it increases the overall time complexity of the tool.

#### 3.2.3.6 Design constraints

Attributes of the overall software design (including data structures, OS features, I/O, and interoperable systems) that constrain the design of this sub-component are presented.

The data structures in this software include two arrays, composed of strings. They will be used to store, determine and display the user's assessment results.

The website assessment will also be used on all types of mobile devices, as long as they have an internet browser. For example, if a Mac user wants to access our website, they will use Safari. If they wanted to use a laptop. Windows will be available as well.

The I/O features include the user selecting answers through radio buttons. The user will also select to move to the next page or move back. The output will be the next section of the questionnaire. After the assessment is 100% completed, the results will be displayed.

The interoperable systems is compatible with other OS and products and devices.

## 3.2 Description for Component n (Help Page)

A detailed description of each software component contained within the architecture is presented. Section 3.2 is repeated for each of the n components.

### 3.2.1 Processing narrative (PSPEC) for component n

A processing narrative for component n is presented.

The user will access the Help Page through clicking the correct button on the navbar at the Home Page. This will display the Help Page filled with frequently asked questions.

### 3.2.2 Component n interface description.

Input: The user will click on the “Help” button located at the top of the Home Page screen on the navigation panel.

Output: The assessment tool will output the Help Page to the user.

### 3.2.3 Sub-Component n.m processing detail

A detailed algorithmic description for each sub-component within the component n is presented. Section 3.2.3 is repeated for each of the m sub-components of component n.

#### 3.2.3.1 Interface description

**Home Page**:

When the webpage is loaded, the user is prompted with a description of the assessor tiny tool. The navbar at the top of the page lists the home button, questionnaire button and the help button. After the description of the tiny tool lays a start button that starts the questionnaire. In the lower right lays the logo of the tiny tool.

**Questionnaire Page**:

The questionnaire page will first describe in a textbox form of what the questionnaire will include. Each section will have three questions for the user to answer. These questions will be inputted with a scale from 1 (being disagreeing) to 5 (being agreeing). The user will be able to go back to the previous section or to the next section using the corresponding button. In the lower middle of the screen will displace the process bar of the questionnaire.

**Confirmation Page**:

When the confirmation page is loaded, the user will be prompted with the summary of their inputs. They will be allowed to edit them if desired by the user. Once the user is happy with their inputs they will be able to click the submit button which will take them the results page.

**Results Page**:

Once the results page is loaded, the user will see the results of their questionnaire. The results will show the users score out of 100 in each section. This score will give the user an idea of which areas of their site they need to strengthen. The user can restart the questionnaire or print the results using the corresponding buttons.

**Help Page**:

The help page, once loaded, will display answers to popular questions for the quality assessment tiny tool. The users will be able to find answers to questions they had about the tool.

### 3.2.3.2 Algorithmic model (e.g., PDL)

The pseudocode listing for sub-component m is presented.

```
    <!-- create the HTML/CSS doc -- >
    <!-- create a section for questions and answers  -->
    <!--create a button to the Home Page -->
    <!-- add a link to the button -->
```

### 3.2.3.3 Restrictions/limitations

*The external environment and/or infrastructure that must exist for sub-component m to operate correctly is provided.*

There aren't any external environment or infrastructure that will affect m to operate correctly.

### 3.2.3.4 Local data structures

The data structures used within sub-component m are presented.

There aren't any data structures that are needed for the creation of a Help Page

### 3.2.3.5 Performance issues

Information on topics that may affect the run-time performance, security, or computational accuracy of this sub-component are presented.

The Help Page can only be launched if the Home Page was accessed successfully. If there were factors such as low internet connection (slow servers), broken links, etc, then the Help Page can't be launched. Having slow servers is a performance issue for the Help Page.

### 3.2.3.6 Design constraints

Attributes of the overall software design (including data structures, OS features, I/O, and interoperable systems) that constrain the design of this sub-component are presented.

The tiny tool will also be used on all types of mobile devices, as long as they have an internet browser. For example, if a Mac user wants to access our website, they will use Safari.

The I/O features include the user clicking on the Help button and the output is the tool launching the Help Page. The interoperable systems is compatible with other OS and products and devices.

## 3.3 Software Interface Description

### 3.3.1 External machine interfaces

For the Quality Assessor Tool, there will not be heavy interaction between other devices per se. The user is free to have a different device (or even tab open) where they compare the questions presented to the website they are assessing, however this isn’t necessary. The product doesn’t require this interaction to function; meaning, the user can simply fill out the questionnaire based on memory. Of course, the tool is in the form of a website, so a computer device is absolutely required to run it, but other than that, there are no external devices needed. In terms of appearance on other devices, the tool should appear the same regardless. The team is inexperienced in terms of websites on mobile devices, but the website should look the same at least on different computers.

### 3.3.2 External system interfaces

The tool relies on an internet connection to connect to the website. Thus, the internet network must be connected to access the product. Otherwise, there will not be any external system that directly will experience the interface of the website. Again, the tool will not parse through any given website, it's a questionnaire about a website. So in the respect of interacting with another product, the tool will not do so. Yet, it may be used to assess that product.

### 3.3.3 Human interface

An overview of any human interfaces to be designed for the software is presented. See Section 4.0 for additional detail.

The human interface of the tool is very important. The entire tool relies on the users input to each question of the assessment to properly judge a website. So, the human interface part is the product itself. The user may also interact with the navigation bar to access the pages of the website. Overall, the initiation and completion of the questionnaire is imperative for the audience to access the Quality Assessor Tool.

## 4.0 User interface design

A description of the user interface design of the software is presented. The Quality Assessor will be made to be used as a website. The UI design will follow that format. The exact number of interfaces cannot be determined yet. Interfaces for every section will be included, as well as, multiple pages on the homescreen.

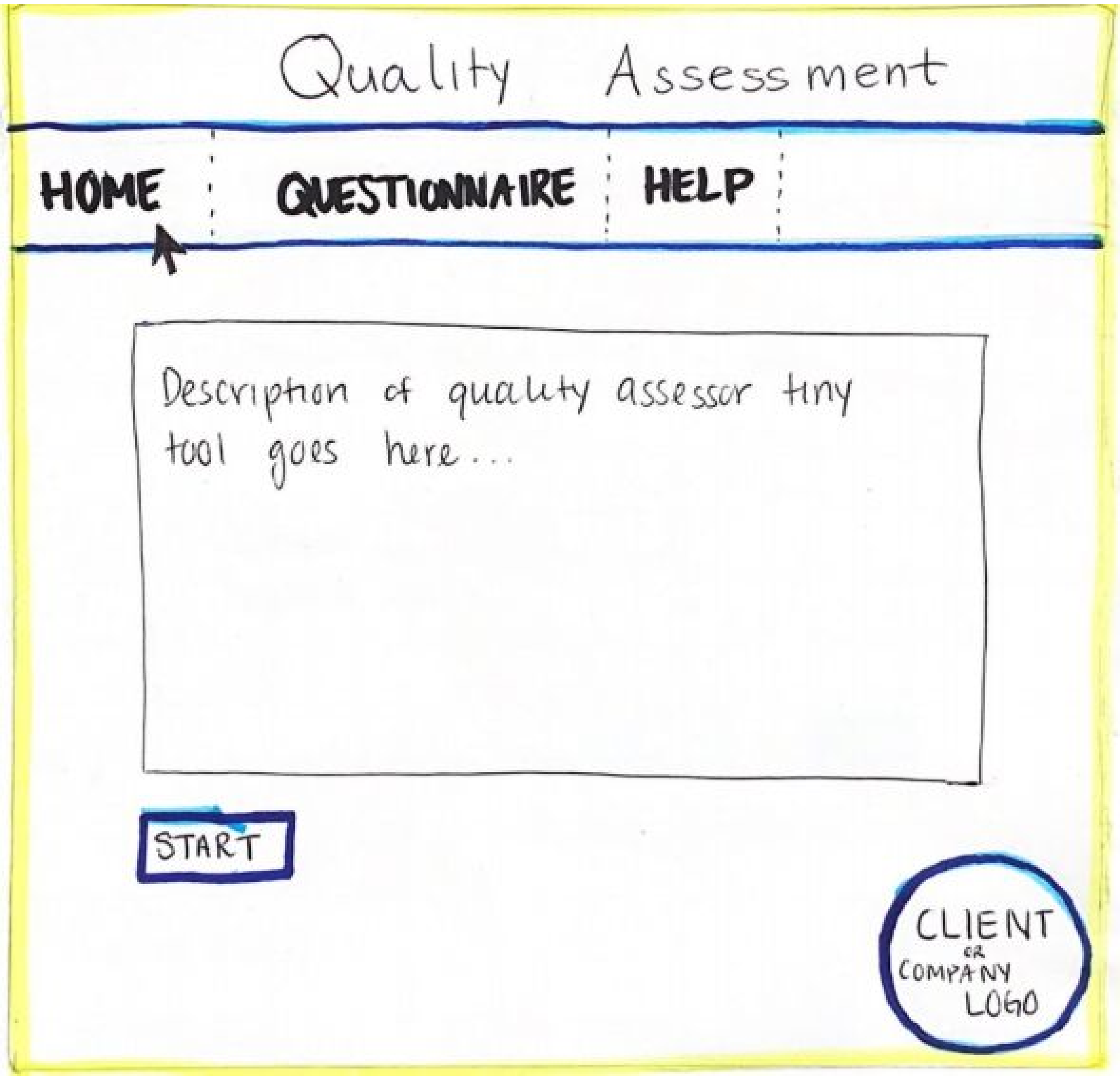
### 4.1 Description of the user interface

A detailed description of user interface including screen images or prototype is presented. Once the user enters the website, the home screen will appear. The home screen will have tabs to access the other pages of the website. The other pages are: questionnaire, and help. Within the questionnaire page there will be a page for each section of questions, a confirmation page for answers, and result page of answers.

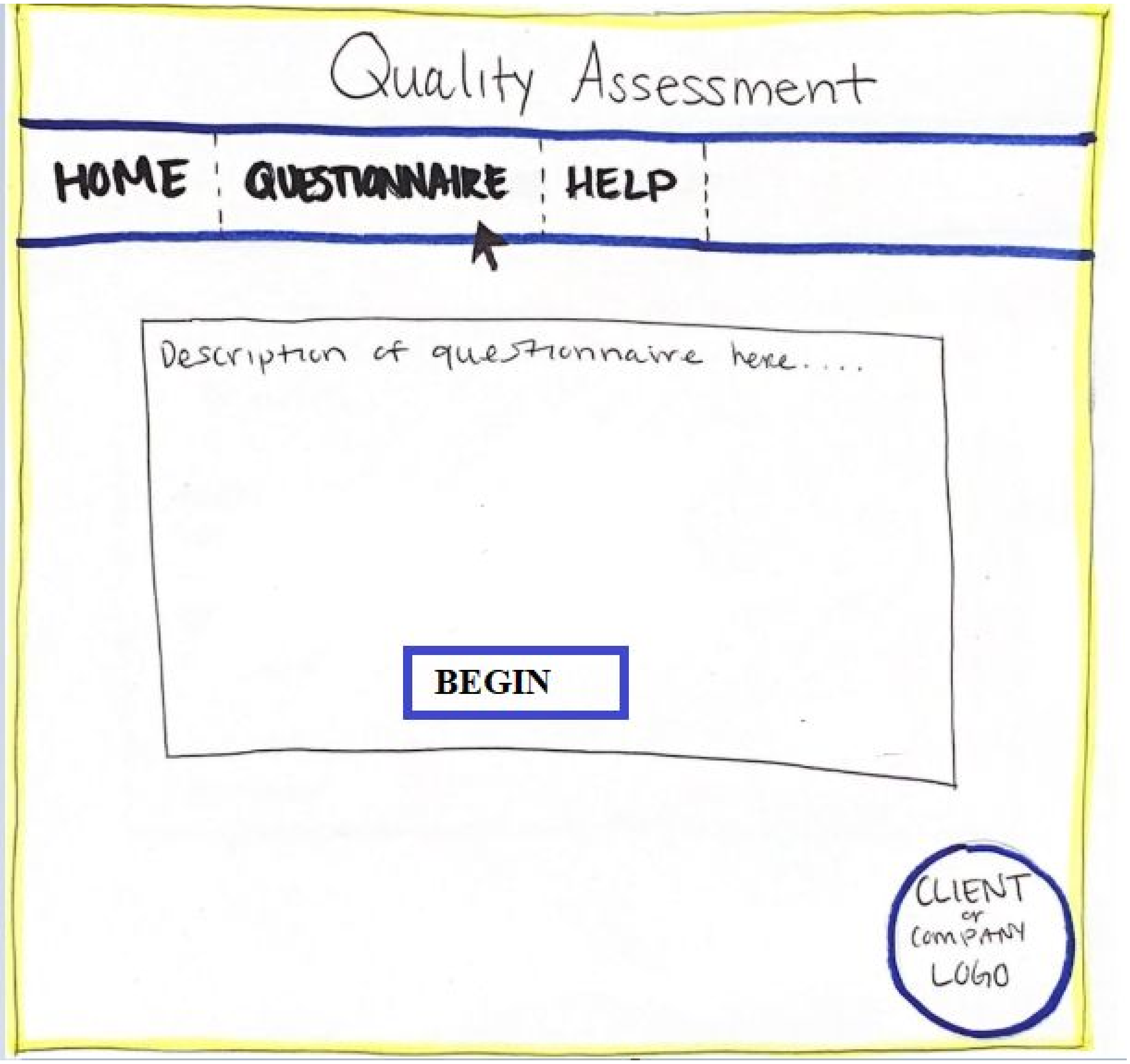
#### 4.1.1 Screen images

Representation of the interface from the user's point of view. These are very rough sketches representing how the UI will look. Some changes will be made due to color choices, button sizing, and lack of JavaScript and HTML knowledge within the group.

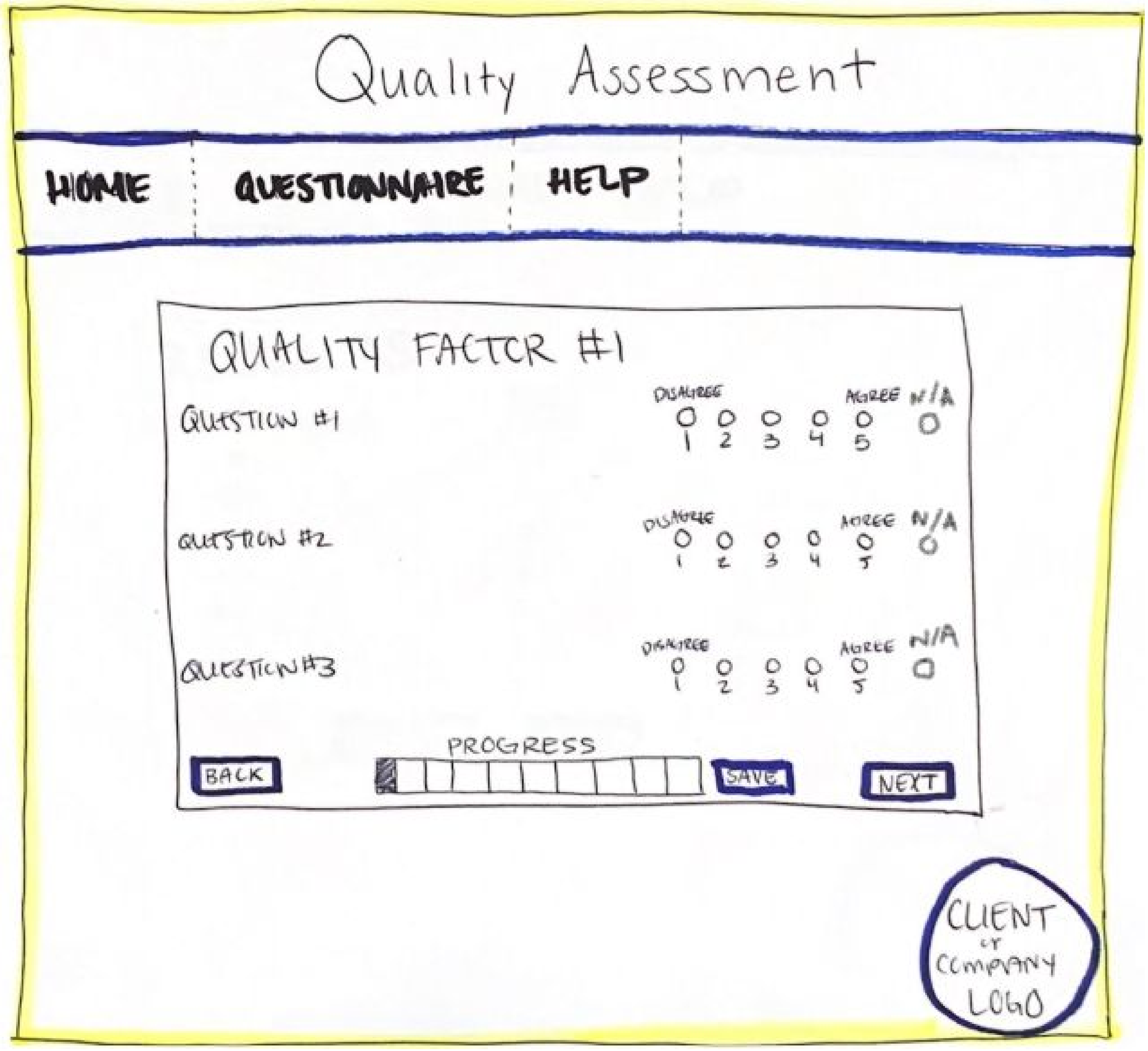
#### 4. 1. 2 Home page Screen



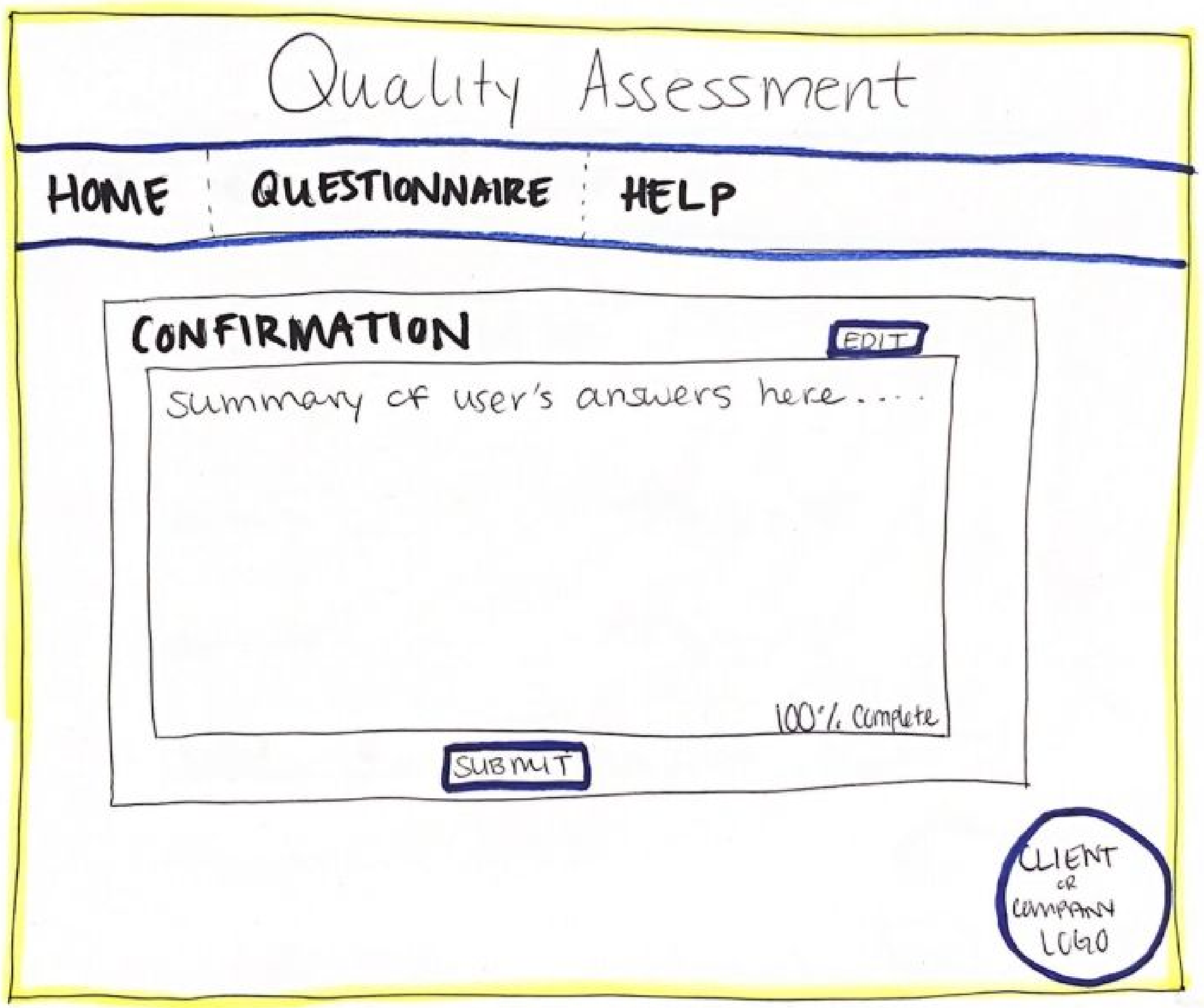
**4. 1. 3 Questionnaire Page**



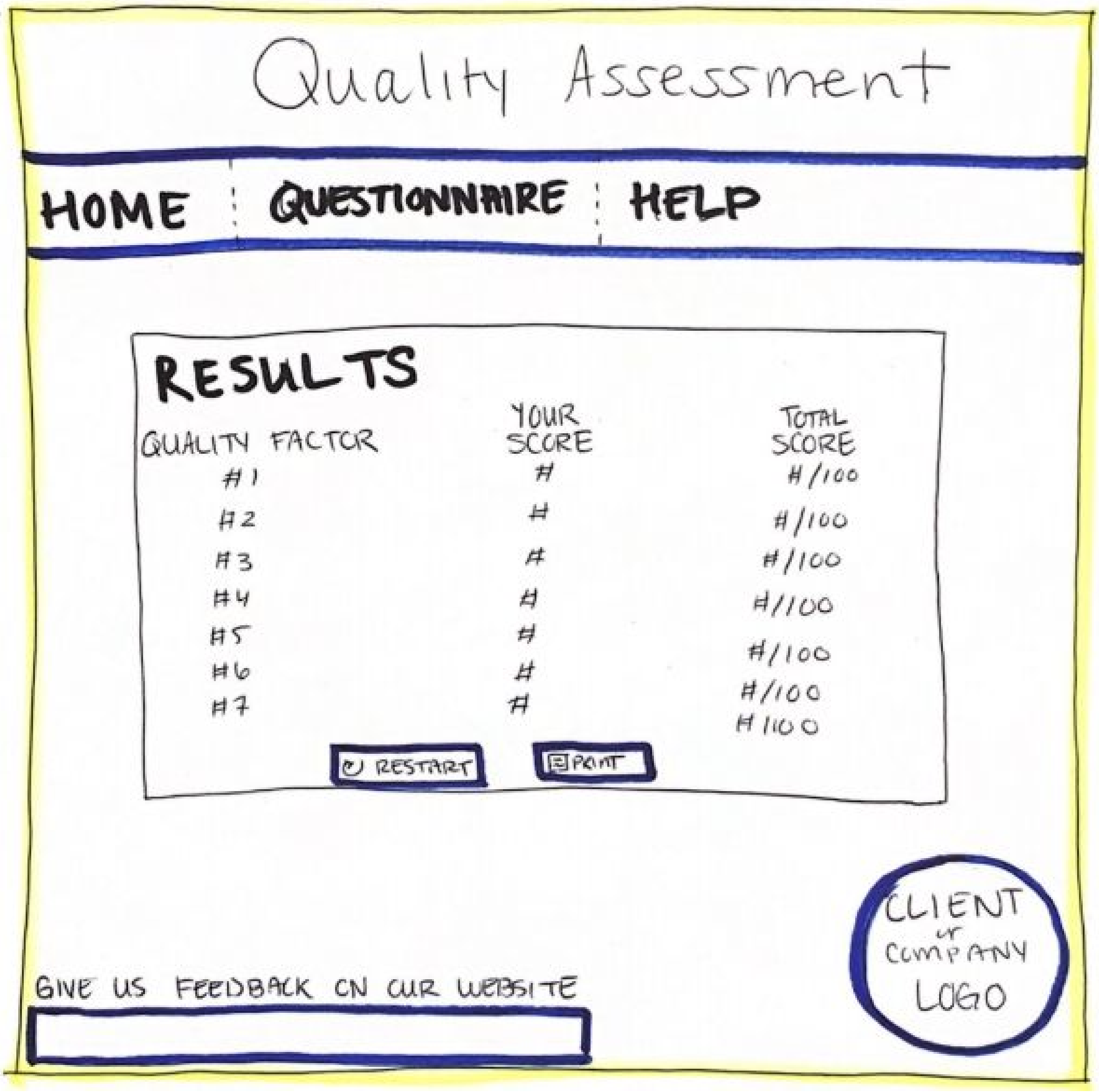
**4. 1. 4 Sample Section page Screen**



**4. 1. 5 Confirmation page Screen**



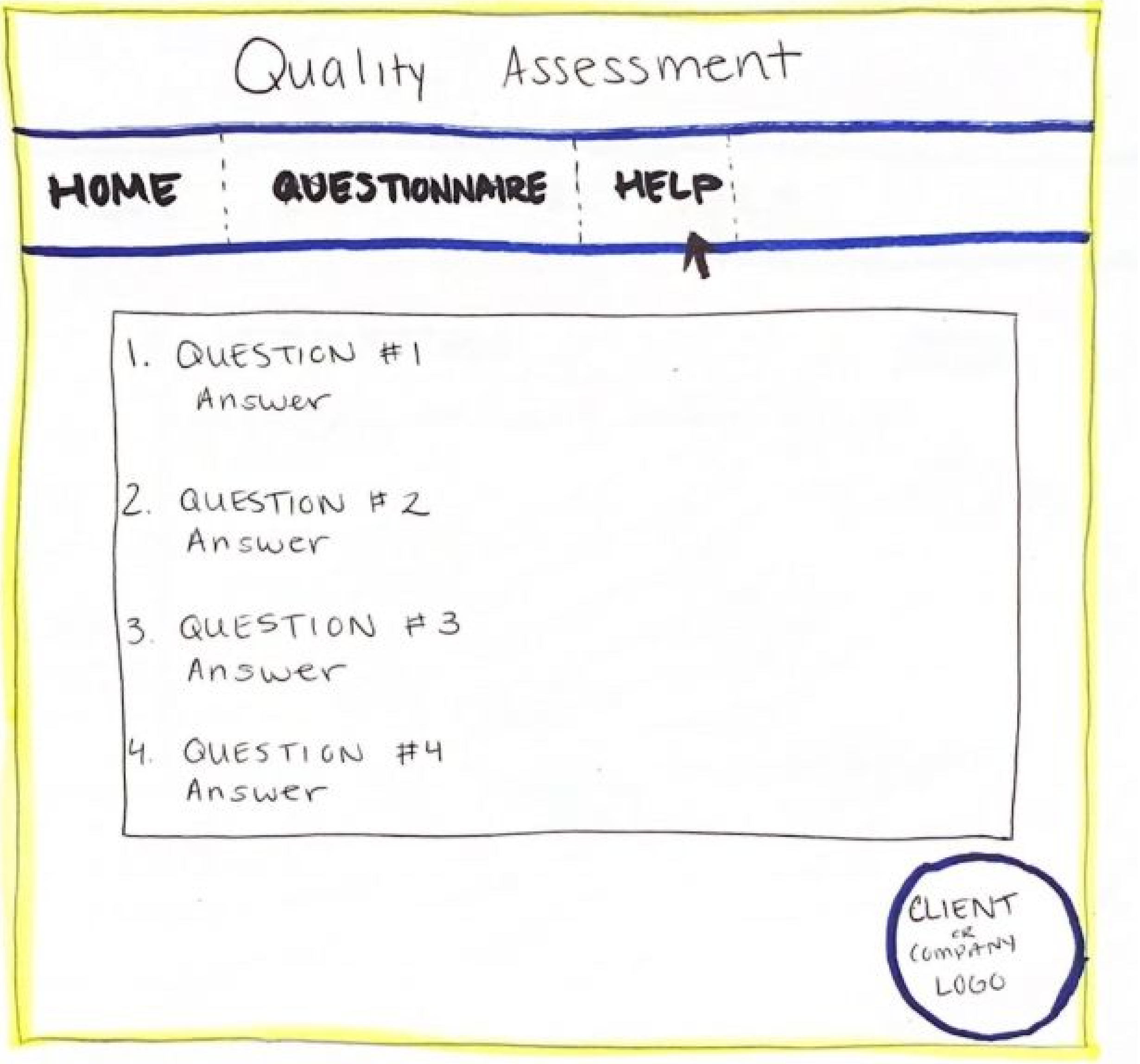
**4. 1. 6 Result page Screen**



RESULTS

| QUALITY FACTOR | YOUR SCORE | TOTAL SCORE |
|---|---|---|
| #1 | # | #/100 |
| #2 | # | #/100 |
| #3 | # | #/100 |
| #4 | # | #/100 |
| #5 | # | #/100 |
| #6 | # | #/100 |
| #7 | # | #/100 |
| | | #/100 |

### 4. 1. 7 Help page Screen



### 4.1.8 Objects and actions

An object that will be on every page of the website is the navigation bar. This object will hold relative links to other pages of the site. The main links to be found on the object for now are: Home, Questionnaire, and Help tabs. The home page will describe the tool and have an option to begin the questionnaire. The questionnaire tab will jump into the assessment itself and begin. The help page will detail any aspect that the user may have questions about.

Home Page

An explanation of the Quality Assessor tool is will be found here. Users can click around the tabs in the upper corner to move through the pages. The logo will be shown of the company or team. A button to start the questionnaire is also displayed.

Start

'Start' button that the user clicks to load up the 'Questionnaire" page. Or else, the user can click the 'Questionnaire' tab.

Questionnaire Page:

An explanation of how the questionnaire is laid out will be presented. The description of the tool is summarized. The logo will be shown of the company or team. A button to begin the questionnaire is displayed. This page loads up when the user clicks the tab ithe home page.

Quality Factor Assessment Page(s)

There will be nine pages that will display questions relating to the quality factor. There will be buttons and a clickable scale to answer these questions.

Back Button:

This object is for users to click a button if they wish to return to the previous page. It will be displayed on the first page of the questionnaire to the last page.

Save Button:

This object is for users to save their selection of answers as they progress through the questionnaire. Users click the button and it will save progress.

Next Button:

This object is for users to click a button if they wish to move forward to the next page. It will be displayed on the first page of the questionnaire to the last page.

Progress Bar:

This object is for users to see their progress as they go through the questionnaire. Every time users click the Next button, the progress bar updates. The progress bar goes up to nine boxes. Each box is meant for each of the nine quality factors.

Scale:

This object has multiple circle buttons that are unfilled. Each question will have five circle buttons. The user will click the button on a scale of 1-5 to fill it. Each scale choice will have a point system. That shows that a question has been answered. A "N/A" option is included.

Confirmation Page:

A summarization of the user's answer are displayed. Each question and the scale is displayed. Buttons to edit and submit the answers are shown.

Submit Button:

The object is for users to click the submit button once they have reviewed their answers. The submit button is displayed in the confirmation page. User clicks it and will be sent to the results page.

Edit Button:

The object is for users to click the edit button if they wish to change their answers.

Results Page:

The results of the questionnaire are shown. The quality factor is shown and then the score the user gave is shown. The overall score is also shown. A color will be associated for when the user is above 50, below 50, or 100. The results will display Low, Medium, or High quality. There are buttons to restart and print the results. There is a textbox to include feedback.

Restart Button:

This object is for users who want to restart the whole questionnaire process. Once clicked, it will take the user back to the Questionnaire page.

Print Button:

This object is for users to select whether they wish to print their results or not. Once clicked, the user's computer settings for a printer will show up.

Feedback Textbox:

This object is for user to type in a response to the question stated. Users will click the textbox for a blinking cursor to appear. A user can enter 250 characters, letters and special characters are allowed.

Help Page:

A list of questions that may be frequently asked will be shown here. The answers for the questions will be shown underneath. The logo will be shown of the company or team. This page loads up when the user clicks the tab ithe home page.

**4.2 Interface design rules**

Conventions and standards used for designing/implementing the user interface are stated.

The team has agreed upon the following conventions and standards for creating the UI for the Quality Assessor Tool:

1. Navigability: The team aims to maximize the user experience in terms of navigation by creating a navigation bar in the header on the website.

2. User Control: The user will also be able to undo what they have previously done by going back to a previous page or restarting the entire assessment. They have the option to edit any of their selections/ratings.
3. The interface is easy to learn for the unconventional technology user
4. The readability of the web quality assessor tool is clear and concise
5. It must be easy to navigate between interfaces
6. Error handling is included, if user does not answer all questions, certain buttons will not work and messages of error are displayed
7. All links and information are consistent all throughout the website tool.

## 4.3 Components available

GUI components available for implementation are noted. Since we are using Visual Studios as our front-end development software, there are a lot of ready-made components that are available for us to use already. The following is a list of controls that we will be using for this software. This is a rough estimate of how many components we will be using.

### 4. 3. 1 Controls

TextBox

A TextBox control, sometimes called an edit field or edit control, displays information entered at design time, entered by the user, or assigned to the control in code at run time.

Label

A Label control is a graphical control you can use to display text that a user can't change directly.

Line

A Line control is a graphical control displayed as a horizontal, vertical, or diagonal line.

Image

Use the Image control to display a graphic. An Image control can display a graphic from a bitmap, icon, or metafile, as well as enhanced metafile, JPEG, or GIF files.

Scrollbars

Scroll bars provide easy navigation through a long list of items or a large amount of information. They can also provide an analog representation of current position. You can use the scroll bar as an input device or indicator of speed or quantity—for example, to control the volume of a computer game or to view the time elapsed in a timed process.

CommandButton

Use a CommandButton control to begin, interrupt, or end a process. When chosen, a CommandButton appears pushed in and so is sometimes called a push button.

Menu

A Menu control displays a custom menu for your application. A menu can include commands, submenus, and separator bars. Each menu you create can have up to four levels of submenus.

CommonDialog

The CommonDialog control provides a standard set of dialog boxes for operations such as opening and saving files, setting print options, and selecting colors and fonts. The control also has the ability to display help by running the Windows Help engine.

ProgressBar

A ProgressBar control provides a window, usually at the bottom of a form, through which an application can display the progress data. The ProgressBar can be divided up into a maximum of ten objects that are contained in a panel collection.

CheckBox

A CheckBox control displays a filled circle when selected; the circle is clear when the CheckBox is cleared. Use this control to give the user a 1-5 scale option. You can use CheckBox controls in groups to display multiple choices from which the user can select one option. You can also set the value of a CheckBox programmatically with the Value property.

Shape

The Shape control is a graphical control displayed as a rectangle, square, oval, circle, rounded rectangle, or rounded square.

## 4.4 UIDS description

The user interface development system is described. The user interface development is done by increments. The tabs of each page will be created first. Then all textboxes and labels will be inserted next. This is necessary for us to know which pages will have what information. From there, we will insert the buttons for each specific page. We can go from there to link the buttons to a specific link so it goes to its desired destination. The company logo will then be uploaded to every page on the assessor tool. The questions for the questionnaire will be added section by section.

### 4. 4. 1 Questionnaire Questions

The following are the list of questions we will pull from to use for the final design:

**Content:**

The content on the page is relevant to the website

The website does not include any spelling mistakes or grammar mistakes

The content is neat and organized on the site

**Function:**

The website has functions that work as intended

The website's response time is under 2 seconds when something is clicked on

The website lets the user know when something does not work properly

**Structure:**

The website has tabs, drop-down menus, and hover links to access other pages.

The user is able to find everything she/he needed

The website is easy to navigate through and consistent.

**Usability:**

Help features are available for users to click on when it is needed

The color themes on the website are appropriate.

The user could find what he/she needed within 2-3 clicks.

**Navigability:**

The website has easy-to-see tabs for users to select different options

Information is easy to find on the site

The website does not have any broken links

**Performance:**

Each page of the website loads in under 2 seconds

The website handles multiple web browsers and settings

Clicking on a button/link launches the correct page all the time

**Compatibility:**

The website is accessible on more than one browser

The website is accessible for a mobile browser view

The website has the ability to change orientations if needed

**Interoperability:**

Is the website accessible on more than one computer?

The website is consistent on multiple browsers.

The website supports multiple operating systems.

**Security:**

The website has a privacy policy

The information entered by the user for the website will be secured

The website deletes your information when you close the browser

## 5.0 Restrictions, limitations, and constraints

Special design issues that impact the design or implementation of the Quality Assessor Tool are noted here.

A huge constraint that the Quality Assessor Tool faces is a novice software development team. Although the team consists of 5 members, nobody has experience with web development at all. Plus, the team dynamic has shifted dramatically with two new team members. Thus, it will be a challenge to not only learn all the languages required to build a website, but it will also be tricky to implement them in a cohesive way as a team. All of this combined will most definitely make timing an additional constraint. To recap, an inexperienced team, new additions to the team, and time are all constraints that the product will need to be completed. All of these factors will surely impact the design of the website as a whole. The inexperience may result in a simple website design and/or layout. The new team dynamic may result in clashing of opinions, or at least different design concepts merging into one on the website. Finally, the most impactful constraint of all, time will ultimately limit what the team can complete in the given period. Again, the team needs to learn multiple languages and integrate them into the Quality Assessor Tool in a very limited time--limiting our end product's design.

## 6.0 Testing Issues

Test strategy and preliminary test case specifications are presented in this section.

### 6.1 Classes of tests

The types of tests to be conducted are specified, including as much detail as is possible at this stage. The emphasis here is on black-box and white-box testing.

For the Quality Assessor Tool, the team plans to complete both black box and white box testing methods.

For black-box testing, the team will have people who have never seen the implementation of the product and have no expectations for what the site will do. From there, the testers will try to navigate the site and give feedback to the team about what was good and what needed improvement from an outsider perspective. This method of testing will really show us what aspects of the prototype will need to improve. It's always beneficial to do black-box testing to see an actual users take on the product at hand.

As for white-box testing, the team will complete some unit tests on each aspect of the pages on the tool. Some initial tests are detailed below. For this type of testing, the developers who are aware of implementation will try to see if the website interacts with the user the way it was intended to. As such, we have expected responses to each test and will adjust the prototype according to what didn't do what it was meant to.

We will be testing many different aspects. We will be testing to see how many people visit the site versus how many people actually take the assessment. We will also be testing how many users begin the test and how many actually finish it all. These are a measure of performance and usability.

The following will be used in black-box testing.

**Load/Enter Site**

**1.1** Invoking the webpage:

Shows that the webpage is accessible

**1.2** Successfully loading webpage:

Shows that the webpage is up and running

**1.3** Navigating webpage:

Shows that the UI is appropriate and that the user is able to navigate it successfully in a timely manner.

**Start Assessment**

**2.1** Begin Assessment button starts pre-assessment:

Pre-assessment give important information that will be used to assist the assessment process.

**2.2** Submit button to start assessment begins assessment:

Shows that the pre-assessment page is easy to understand.

**Questionnaire Sections**

**3.1** Statement questions are listed for user:

As the questions are critical to the performance of the tiny tool. The questions asked have been meticulously selected. Interaction with each question will prove this.

**3.2** Selection from scale of 1-5 and N/A is available to answer questions:

An answer scale will allow the user to get the most accurate results from our tiny tool.

**3.3** Assessment Progress is shown:

The progress bar is a visual aid to help the user understand how long the assessment is.

**3.4** Back button is clickable:

Allowing the user to go back to the previous section gives the user multiple ways to interact with the site. Making it an important feature to test.

**3.5** Next button is clickable

After completing a section the user will be allowed to move onto the next section. This is critical to the success of the website.

**Confirmation Page**

**4.1** Allow use to edit selections:

Users make mistakes, this shows that the user is able to edit their selections before getting their final results.

**4.2** Submit button is clickable:

This is the last chance the user will have to edit their selection before the final results will be generated.

**4.3** Shows summary of user's answers:

Shows the user everything they have selection in one area so the user may review their selections.

**Results Page**

**5.1** Results are calculated and shown correctly:

The final results are shown to the user

**5.2** Print button is clickable to allow user to print results:

This method is the easiest and most effective way to allow the user to save their results to use to better their own site.

**5.3** Restart button is clickable

Allowing the user to restart the test is an important feature to keep users on the site, enjoying assessing sites

**5.4** Feedback on quality of website is given.

Feedback is important, without it the site wouldn't be successful. This feature is critical to the website.

**6.2 Expected software response**

**Load/Enter Site**

The user has entered the website and the webpages have all been successfully loaded.

**Start Assessment**

Begins pre-assessment. Once users have read the description of the questionnaire, users will be allowed to begin assessment by clicking the begin button.

**Questionnaire Sections**

The user will go through nine sections to complete the Q&A. Each section the user will be given a scale from 1 to 5 (1 being strongly disagree and 5 being strongly agree) to selection from. Once all questions are answered the user can go to the next section. The user can also go back a section with the back button.

**Confirmation Page**

The user will review their answers and edit them as needed. A submit button and an edit button will take them to the next or previous section.

**Results Page**

Will display correct results and allow user to print the page as needed. The user will see the feedback on the quality of their website. A restart button will allow the user to restart the quiz.

### 6.3 Performance bounds

To perform at all, the Quality Assessor Tool will need an internet connection. Thus, if the internet connection is weak, the product will not carry out as quickly as one would like. In addition to this, the tool will be tested and implemented through computer devices. So, the product may not be functional as expected on other devices.

### 6.4 Identification of critical components

Those components that are critical and demand particular attention during testing are identified.

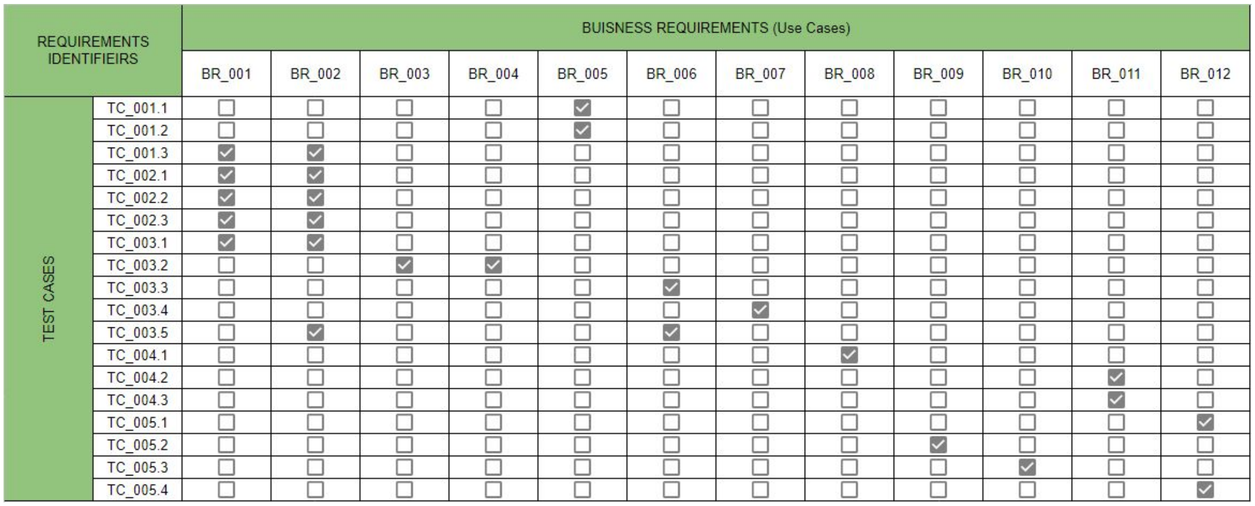
Perhaps the most critical component of our tool is the computation required to compile the feedback score. While simply summing the results of the questionnaire may seem easy, it must be thoroughly tested. The reason for this is that it's the entire purpose of the product. The user’s goal is to complete the assessment to get feedback on the given website. If this requirement is not functional then there is little purpose to the tool.

Additionally, its relatively important to ensure that there are no dead links on our website. It would be a shame to develop a tool that judges other websites, when our own website doesn’t adhere to the same guidelines. So, the navigation bar in particular needs to be clear and concise with respect to this.

# 7.0 Appendices

## 7.1 Requirements traceability matrix

Each requirement and the respective use case that completes it is displayed.



| REQUIREMENTS IDENTIFIEIRS | | BUISNESS REQUIREMENTS (Use Cases) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | BR_001 | BR_002 | BR_003 | BR_004 | BR_005 | BR_006 | BR_007 | BR_008 | BR_009 | BR_010 | BR_011 | BR_012 |
| TEST CASES | TC_001.1 | □ | □ | □ | □ | ☑ | □ | □ | □ | □ | □ | □ | □ |
| | TC_001.2 | □ | □ | □ | □ | ☑ | □ | □ | □ | □ | □ | □ | □ |
| | TC_001.3 | ☑ | ☑ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | TC_002.1 | ☑ | ☑ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | TC_002.2 | ☑ | ☑ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | TC_002.3 | ☑ | ☑ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | TC_003.1 | ☑ | ☑ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | TC_003.2 | □ | □ | ☑ | ☑ | □ | □ | □ | □ | □ | □ | □ | □ |
| | TC_003.3 | □ | □ | □ | □ | □ | ☑ | □ | □ | □ | □ | □ | □ |
| | TC_003.4 | □ | □ | □ | □ | □ | □ | ☑ | □ | □ | □ | □ | □ |
| | TC_003.5 | □ | ☑ | □ | □ | □ | ☑ | □ | □ | □ | □ | □ | □ |
| | TC_004.1 | □ | □ | □ | □ | □ | □ | □ | ☑ | □ | □ | □ | □ |
| | TC_004.2 | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ☑ | □ |
| | TC_004.3 | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ☑ | □ |
| | TC_005.1 | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ☑ |
| | TC_005.2 | □ | □ | □ | □ | □ | □ | □ | □ | ☑ | □ | □ | □ |
| | TC_005.3 | □ | □ | □ | □ | □ | □ | □ | □ | □ | ☑ | □ | □ |
| | TC_005.4 | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ☑ |

## 7.2 Packaging and installation issues

Special considerations for software packaging and installation are presented.

The Quality Assessor Tool will be packaged in the form of a website created on Google Chrome. Thus, for the best representation of the product, using this browser is highly recommended for the best experience. Otherwise, there will be no considerations for installation (because its a website) nor packaging. The website tool will work on multiple web platforms. But it will be recommended to use Google Chrome for the best results. In addition to this, the host of our website will be the University of Michigan. So, some users may be restricted in terms of viewing the website.

## 7.3 Design metrics to be used

One design metric we will be using is Goal-Oriented Software Measurement. In this design metric, there is a specific, measurable goal that is needed to be set using measurable data. We will then create a list of questions that will help achieve the goal.

We will be using this, where our goal is to help users identify great quality websites. We will be creating and using a questionnaire to determine whether or not a website is good quality. Another design metric we will use is architectural design. In this, we will be using a similar technique to black box testing. Lastly, we will be utilizing interface design metrics. With this, we hope to test the functionality and design of the interface. This will be used to determine between a good, user friendly interface, vs a bad one.

### 7.4 Supplementary information

Works Cited

Duck, Roast. *Layered Architecture Style*, www.student.cs.uwaterloo.ca/~cs446/1171/Arch_Design_Activity/Layered.pdf.

“JavaScript HTML DOM.” *w3schools.Com*, https://www.w3schools.com/js/js_htmldom.asp.

Kulkarni, Aditi. “Design Metrics for Better Design Decisions.” *Medium*, UX Collective, 24 May 2017, uxdesign.cc/design-metrics-for-better-design-decisions-fb6601b45d04.

Pressman, Roger S., and Bruce R. Maxim. *Software Engineering: a Practitioner's Approach*. McGraw-Hill Education, 2020.

Rovers, Cyber. “WMITS Software Design Specification” https://www.luminpdf.com/viewer/5dcd7c6be105400013efb33a