UAT Test Plan for Morse web learning project

1. Scope

1.1. Objectives and business requirements

In this section, outline the business requirements. In other words:
- The goal of this user acceptance test is to figure out whether a user can use the morse learning website without errors.
- Success of this UAT process will be measured by percentage of test cases passed. Expectation is for 100% pass of all test cases.
- We will measure our success by the amount of users on the site monthly. Ideally we would like several hundred users monthly. *- this would be the success of the project - not success of UAT. Should clarify exactly what this section is for.*

1.2. Scope

For this UAT test we would like to;
- Verify that the website content is downloading and displaying properly includiing all graphics, text and audio,
- Identify any problems users have with the hyperlinks and anchor tags, ensuring users are connected to the correct page and desired location.
- Get feedback from users on any errors or unexpected results.

2. Testing team

In this section, list out members of your QA team and what their roles will be during UAT.

| Name	
Adam Tester	UAT coordination -including communication between end users and QA team Writing UAT report
Boris Tester	Design test cases - end to end website function Conduct training with end users
Charlie Test	Set up stage environment and test data creation as required

3. Milestones and deliverables

This section contains all deliverables for successful UAT execution.
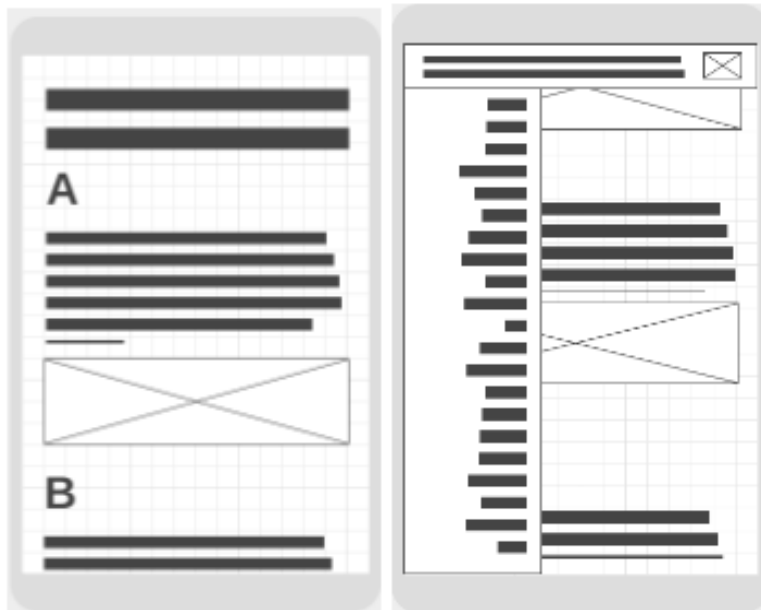
3.1. Design & testing process

In this subsection, share design & wireframes to ensure the whole QA team is on the same page. Then, describe how and when the testing will take place

SITE WIREFRAME: WEBSITE
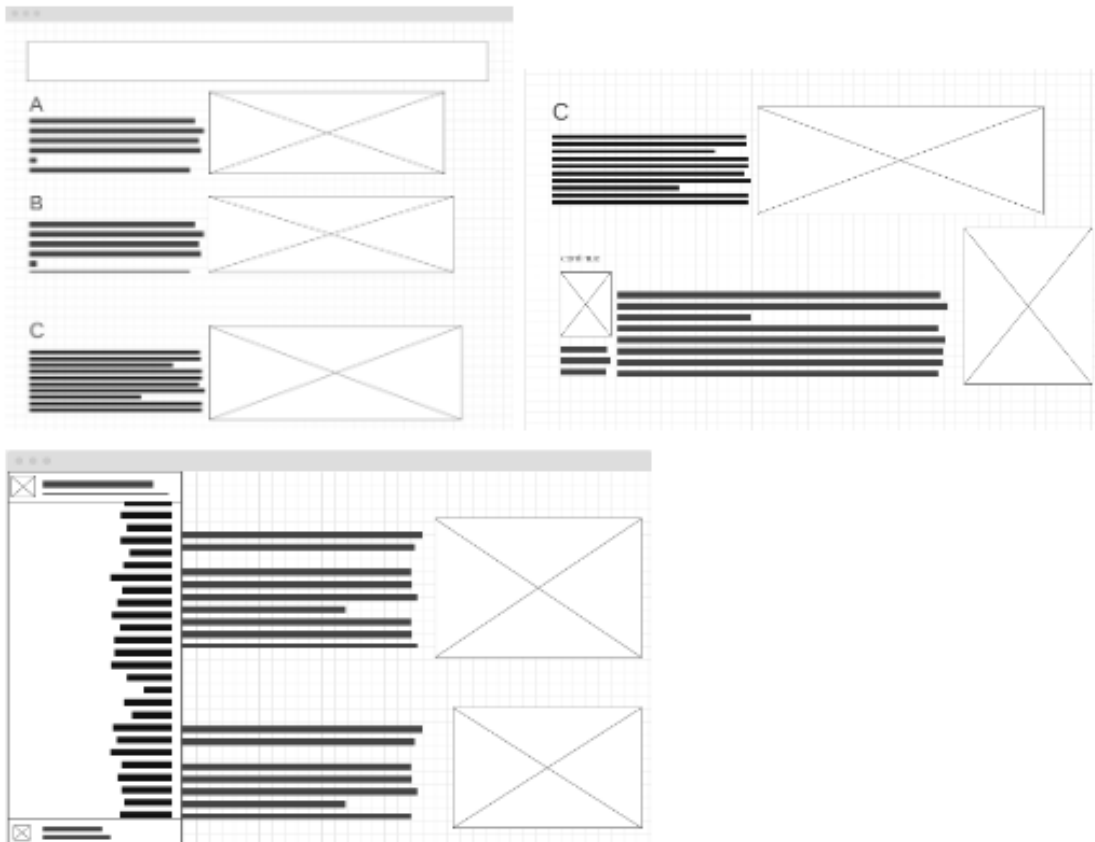
SITE WIREFRAME: MOBILE / TABLET

These wireframes depict the website on both phone and computer. The side bar that is shown will be able to slide back into the side of the computer when using the website. The sidebar will display anchor text that will send them to their desired web page off of this side bar. The search bar will also allow users to find other pages just as the sidebar did.
PHONE

The testing will occur in 4 steps
1. **Preparation** Staging environment and test data set up: Environment set up by Charlie Tester. This environment will closely mirror production. Create a snapshot of the production database - Charlie Tester
2. **Training**: UAT end user testers will be trained by Adam Tester. We will hold face to face training with UAT testers in the last couple days of october
3. **UAT Execution**: test cases and have our testers/ reporters on said test cases and have our testers/reporters on said test cases.
4. **Reporting**: full data analysis, bug triage, and meeting on what remains to be done

Please go through the different designs and notes on documents for a reminder of how this feature works..

Deadline for design & testing process: Jan 20, 2023.

3.2.   Staging environment

Describe requirements for the staging environment—this will typically be company-specific, but it should be as close to production as possible.

The requirements for the staging environment include making a copy of the website to and make sure its not live so they do not break thing live.

Deadline for staging environment february 5

Example:
Our staging environment will be accessible for all UAT testers on uat.acme.com.

Make a copy of the production database, and onboard users via their usual profile—**tell them to double check the URL to make sure they are not breaking anything on live.**

Deadline for staging environment: Jan 27, 2023.

3.3.   Training

In this section, go over how you will proceed for training beta testers.

We will hold a UAT meeting in the last few weeks of october

Well have john set up that meeting and walk through what the new features does and how to make the most out of it

The meeting will last 10min - present the website features and business objectives

Example:
We will be holding UAT meetings the first few weeks of February.

We'll have [name] set up those meetings and walk them through what the new feature does and how to make the most out of it.

First meeting - 30 minutes - present new feature & business objectives
Second meeting - 1 hour - how to log to staging environment, enable and best practices on the new feature
Third meeting - 1 hour - how to report on test cases

Deadline for training: Feb 15, 2023.

3.4.   UAT Execution

Describe how and when UAT execution will take place—from onboarding to having testers report on test cases.

The UAT execution will take place in 1 week. During these executions we will need to ensure that everything on the website is tested fully.

Steps
1) Onboarding. Onboard each tester individually, help them set up on staging, and explain what we expect of them (briefly touched on during training as well).
2) Test case execution. Each tester will be given specific test cases (see below), and report bugs and feedback via the Marker.io widget
3) Once done, record a quick meeting with the testers to get feedback on the experience we can come back to during the QA meeting.

Deadline for UAT execution: October 28th , 2029

3.5.   Reporting & data analysis

The UAT report provides for a full analysis of individual test cases, to understand what testers struggled with, any functional bugs as well as general feedback and areas of improvement.

Deadline for completion of the UAT report November 5th 2029

4.   Environmental requirements
4.1.   Hardware requirements

Some software (design, video editing…) can be demanding on hardware specifications.

- 1.9GHz or faster processor. Quad-core or better recommended.
- 1.5 GB of RAM; 8 GB of RAM recommended.
- Hard disk space: Minimum of 1 GB, depending on features installed.
- Hard disk speed: to improve performance, install on a solid state drive (SSD).
- Video card that supports a minimum display resolution of 720p.

If that is the case, outline the minimal and recommended requirements so the QA team can verify that the software runs on the testers' machines.

4.2. Software requirements

If any extra software or dependencies must be downloaded and installed, list them here.
(nil required - web based functionality only)

5. Features to be tested

It is crucial that both the QA team and the testers know what features must be tested, especially if you're testing a lot at once.

1) Test that the content of each section downloads as expected,
2) Test that all audible links are able to be heard and understood as expected
3) Test that all graphics download and display as expected.
4) Test all links within the website
5) Test sidebar to make sure in works probably

Without this, it's too easy to get sidetracked, and lose time or valuable data from your testers.

5.1. Feature 1
5.1.1. Pass/fail criteria

Add a clear description of what the pass and fail criteria is for each feature.

1) Links : if the link is able to take you to the right page or section of the page, than it would count as a pass but if it takes you to the wrong page or does not work at all than its a fail
2) Graphics: if the graphics appear as expected then pass, If graphics do not download and render as expected the test should fail
3) Text :appear and makes sense - pass, if it does not appear - fail.
4) Text: grammatical errors or language change suggestions - flag as possible improvement.
5) Audible files - must be able to be heard and understandable at a speaker level of 70% to pass, otherwise fail.
6) Audible files must be able to be heard and understandable using headsets at a maximum level of 70% to pass, otherwise - fail

●

5.1.2. Test cases

Write step-by-step, detailed but concise instructions on how to test the feature.

1) Go into website
2) Click on any and all links

3) Go to different pages to test links on those pages
4) Test sidebar if work or not

5.2. Features to avoid testing

Avoid testers being sidetracked by specifying what features must be avoided during testing.

Nil. This UAT process is intended to test all features end to end.

This is particularly relevant if you're testing a lot of features at once, or if your software is complex enough that testers might not recognize that they're testing the wrong feature.