UAT Test Plan for E-ink display Client: Anthony Edwards

1. Scope

1.1. Objectives and business requirements

In this section, outline the business requirements. In other words:

- Our goal with this project was to limit the need of having to check your phone when you want to see the weather forecast today or the next days of events on your calendar. We hope to accomplish an E-ink display that will act as your new calendar and weather forecast stopping the need for normal basic calendars instead of looking through all those dates you will be able to glance at the E-ink display and see the weather and your events. Instead of taking out your phone and looking or looking through the millions of things you've written on your calendar.

- We will measure success by how well our implementations work and how useful they are. In addition to this, we will also judge it by convenience. As that is one of the main goals. A convenient, easy-to-read calendar and event screen.

The goal of this user acceptance test is to figure out whether our Web app for the E-ink display is easy to use.

1.2. Scope

In this section, outline the scope. This means:

- The main thing we are trying to fix is people who are busy and have all their events digitally can now look at their wall on the way out of their house at their E-ink display and they will be able to see what they have for that day and how the weather will be for them that day. There is also a Webapp to their convenience that they can have a quick look at and they can get all the information right there for them
- We are testing the usability of the web app to see the convenience for online users to see how easy it is to access and navigate and to do what they want to do and find out whether it is hard for them to navigate the web app if so then we will have to think about adjusting the UI.

For this UAT test, we'd like to:

- See how easy users find the web app to navigate
- Make sure the calendar integration is correct.
- Make sure when the weather is updated it changes to the correct weather pattern

2. Testing team

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

Name	**Responsibilities**
Sam	UAT Coordinator - handles communication between end users and the QA team
Tom	Set up staging + usability test cases
Luke	Design test cases for the users

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.

The designs should be ready at this point, so this stage is just a matter of keeping everyone aligned—and acknowledging what needs to be done.

- https://github.com/SSkimmd/COMP2003/tree/master/Design%20Documents

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

The testing will occur in 4 stages:

1. Staging environment: set up by Tom, this environment should closely mirror production. Create a snapshot of the production database.
2. Training: UAT testers will be trained by Sam. We're holding UAT meetings in the first few weeks of February.
3. UAT execution: create test cases and have our testers/reporters report on said test cases.
4. Reporting: full data analysis, bug triage, and meeting on what remains to be done.

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.

Our staging environment will be set as a branch on Github so it won't mess up the current working build and can be safely tested without worry of any major breaks or any alterations to the live build

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 be holding UAT meetings in the last weeks of January to the first week of February

We'll have Luke + Tom 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 the 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.

Execution will take 3 days. During these, we get multiple people who have never used the web app before and we will see how they navigate through the GUI and see if they can complete some simple tasks

Steps:

1) Onboarding. Onboard each user individually, give them some set tasks, leave them without any help, and see how long it takes for them to do these tasks this will give us a valid user usability test which would allow us to see how good the current GUI is
2) Depending on how the results go we will either add new GUI and new tasks or if the times take longer than expected we will update the GUI and make it easier and more user-friendly to navigate.
3) Once per iteration we will talk with the users and see what they think, if they found it easy any improvements, and see what ideas they have and see how viable they are.

Deadline for UAT execution: Feb 19, 2023.

3.5. Reporting & data analysis

Full analysis of individual test cases—understand what testers struggled with, what the general feedback is, and areas of improvement.

User testers found it easy to navigate and didn't have issues going page to page and understanding what to do which resulted in our test being a success in making it simple to understand and user-friendly. Areas of improvement were updating the UI slightly and making some slight altercations which were made.

Deadline for reporting & data analysis: Feb 23, 2023.

4. Environmental requirements

4.1. Hardware requirements

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

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.

minimum requirements

- **Memory:** 4 GB
- **Graphics Card:** NVIDIA GeForce GTX 960
- **CPU:** Intel Core 2 Duo E8400
- **File Size:** 900mb
- **OS:** windows 7+

recommended specs

- **Memory:** 4 GB+
- **Graphics Card:** NVIDIA GeForce GTX 1070 +
- **CPU:** Intel Core 2 Duo E8400 +
- **File Size:** 1GB
- **OS:** windows 7+
-

4.2. Software requirements

If any extra software or dependencies must be downloaded and installed, list them here.

Python3.9
Requirements.txt provided
Powershell

5. Features to be tested

This section is more important than it seems—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.

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

5.1. Feature 1

Weather API display.

5.1.1. Pass/fail criteria

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

- **Pass:** Weather API displays the correct data for the area selected.
- **Fail:** Weather API displays incorrect data or no data for the area selected.

5.1.2. Test cases

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

1) Login to the page
2) Click the weather box
3) The weather box opens.
4) Change location to the place you desire
5) Click refresh
6) Compare results to Google weather

5.2. Feature 2

Calendar

5.2.1. Pass/fail criteria

Pass: Calendar displays the correct data for the calendar selected.
Fail: Calendar displays incorrect data or no data for the area selected.

5.2.2. Test cases

1) Login to the page
2) Click calendar box
3) Calendar box opens
4) Check calendar
5) Compare to Google calendar

5.3. Features to avoid testing

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

5.3.1. Feature 3

Device Connected Box

5.3.2. Feature 4

Customisation Box

6. Signoff

I hereby accept this final product. (Yes/No)

(Signature)

Client Name, Position, Organisation
Date:

Anthony Edwards

Associate Lecturer