**Pilot Support and The Process of QA**

Employee Manager v1.0

Take some time again to explore this site to understand it better.

Now think back to the previous conversation, take some time to answer these questions:

1. Was the customer justified with their complaints?

Yes, because it seems that the customer was not aware of the roll-out being a prototype.

1. Did the QA do a good job responding to the customer’s complaints?

Initially yes, but at the conclusion the QA didn’t ask the proper questions to gain the necessary clarification.

1. Would you address them in a different way?

Yes, by ensuring through communication that the roll-out was a prototype and not production. Would not have made feature promises or set feature expectations with the customer.

1. What information would you need to adequately respond to their concerns?

Documentation of the customer’s experience and concerns.

Based on the conversation above, create a checklist of features that the customer would like the website to do.

* A better presentation esthetically,
* For the Save functionality to work,
* Correct employee data,
* Addition of the Email field,
* The ability to search,
* The ability to log-in/out.

Try to imagine yourself as a customer in this situation, what features would you want this employee manager site to have?

* For end-users to be informed of the proper release phase,
* The purpose intent for the overall functionality of the application,
* Clarity on how the information is to be displayed,
* Clearly stated labeling within the application,
* Intuitive navigation of the application,
* An Add button,
* A delete button,
* Buttons that are responsive (regarding the Save and Cancel buttons).

Now that you’ve thought about things from the customer perspective, let’s shift back into the QA mindset.

Here’s some documentation for Employee Manager: <https://devmountain-qa.github.io/employee-manager/1.0_README.html>

Let’s remember what the Process of QA is:

1. Planning
2. Testing
3. Reporting
4. Following Up

Use the documentation above to create a document that answers these questions:

1. What are you testing?

Employee Management Application (Front-end)

1. How long will you test for?

The actual duration is unknow. All of the documentation is not available (no access to JIRA) to properly measure the testing duration. Within the prototype phase and test iteration, the “Current Desired Functionality” need to be tested. While the production application is being developed the “Future Functionality” need to be tested.

1. Are there pieces you aren’t going to test?

No, because both the current desire functionality and future functionality need to be tested.

1. How do you plan on testing?

* Ensure that there is access to all the documentation.
* Once access of all the documentation has been gained, review all the documentation to determine the testing focus.
* Conduct exploratory testing to become familiar with the Employee Management Application.
* Review the user acceptance criteria.
* Prioritize the features identified for testing.
* Document the prioritized features in a test plan.

1. How do you know when you’re done testing?

In this case, it’s difficult to know because access to all of the documentation is not available.

In general, the testing may be identified as being done when all the test cases (one for each of the features) that have been targeted for the test cycle/sprint have successfully passed.

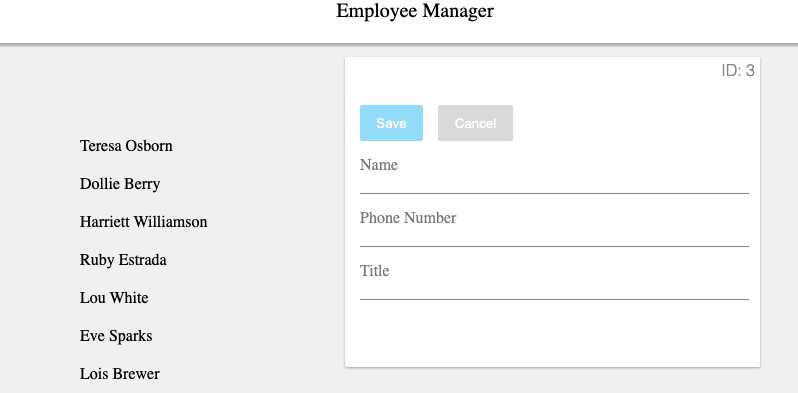
By answering these questions, you’re starting to create a plan for your testing, the first step of the Process of QA!

Now you’ll want to use that plan and take the next 30 minutes to do some testing! Take a close look at each feature in the application! This is the next phase, the actual testing!

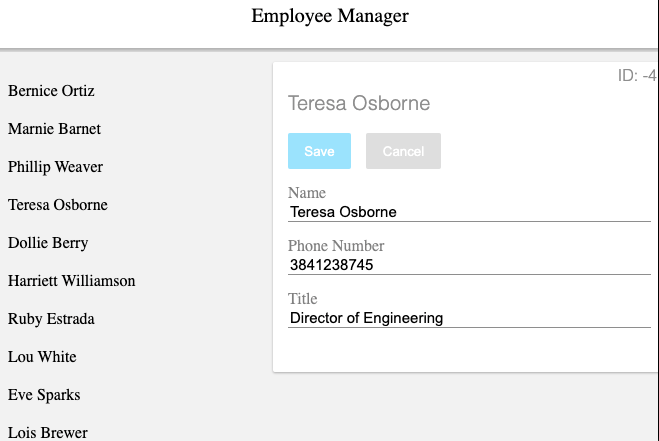
Now based on your findings from running some tests, create a report with a list of all the issues/bug you found, but also include the positives! If you think the application is a good start, be sure to say that!

**Issues/Bugs:**

* No documentation/instructions for roll-out and version.
* The user can only select the cancel button before the save button is selected, if a change has been made.
* The user can remove the Name, Phone Number, and Title. Then select Save and the record to deleted until the page is refreshed:



* Negative ID Number (-4 in the top right corner) :



**Positive:**

* The application is a good start with the intent to keep tracking of an Employee list. But it is vital to effectively communicate with the end-users before rolling-out any phase of the application.

And our last phase, following up. Take the next 30 minutes to write out a report to the customer from earlier in the lab. Include the issues you’ve found, and how you’d plan to address their questions and concerns.

**Report:**

As a follow-up on the conversation held with an end-user, regarding the functionality of the recent release of the Employee Manager application, here are the concerns that were captured:

* A better presentation esthetically,
* For the Save functionality to work,
* Correct data (The employee’s Name, Phone Number, and Title),
* Addition of the Email field,
* The ability to search,
* The ability to log-in/out.

During our conversation, the application phase of prototype and not production was conveyed for the current roll-out.

The following are additional issues/bugs that were identified by the QA team:

* No documentation/instructions for the roll-out and version.
* The user can only select the cancel button before the save button is selected, if a change has been made.
* The user can remove the Name, Phone Number, and Title. Then select Save and the record to deleted until the page is refreshed.
* Negative ID Number for an employee (-4 in the top right corner).

Per the issue/bugs found by the QA team, here are the next steps:

* Create a plan that includes an overview, test criteria, entry criteria, exit criteria, environment details, as well as any other details. In this test plan each finding will be documented.
* Create a test plan for each finding. The test plan will be inclusive of the description of the finding, the steps required, post condition, and documentation of results.
* The necessary bug reports will be created and include a description, steps to reproduce, the expected results, a documentation of the actual results, and environment tested within.