Maintenance Request Analysis and Design

“The Simplifiers”

Reagan Leonard

April 8, 2020

This document is a summary of my individual design critique of Group 9’s wireframes for their app to assist in the task of interacting with a maintenance request/service system. Contained in this document is an introduction to their user interface as well as an explanation of any terms needed to better understand their wireframes or my critique of them. Most importantly, I have created nine UARs for their wireframes, seven of which are related to problems within their design. I then provide a conclusion summarizing these problems and a reflection on my findings. The following is an outline of the contents of this document:

Executive Summary

The following document is a summary design critique of the wireframes created by Group 9 for an app to assist in the task of interacting with a maintenance request/service system. I surveyed these wireframes in detail to find what was good and--more importantly for this assignment--what was bad about the design of the system. In general, it was honestly difficult to find many different problems with the design. The group seems to have thought out their system very well and how the user would interact with it. However, I found a major problem with the overall design of the interface being much too small for the mobile device that it was designed for. I also found issues with the system relating to 6 of Nielsen's 10 design heuristics. These are outlined in the UARs that follow and are summarized in the Conclusion section. For a detailed description of each of the design issues I found and their corresponding solution, see the UAR associated with it.

Introduction

This system is meant to be an interface between residents of a community and maintenance workers who perform maintenance work on behalf of those residents. It also has a "supervisor/Manager" user type, but does not go into much detail on that. The system allows each of these types of user to sign-in and perform a number of actions that ideally result in a maintenance problem being resolved or fixed in a timely and efficient manner. The system is intended to make the communication of these maintenance issues to the maintenance workers easier and more direct, as well as communication between the workers and the residents regarding further details of the issue if necessary. Either the workers or the residents may initiate a direct message (DM) with the other party if any clarification questions or concerns need to be voiced at any time before or after the maintenance visit.

Another main purpose of this system is for workers to be able to report on the status of a current or finished job. This is for transparency to both their supervisor/manager and to the resident. All parties involved are able to view the status of a job which can include "in progress", "paused", or completed". The resident also receives a notification in the app whenever the status of an ongoing request is changed from any one of these statuses to another. Some example notifications might be "John has accepted your request ID #41925" followed by "John has set your request ID #41925 to ACTIVE" followed by "John has marked your request ID #41925 as COMPLETED".

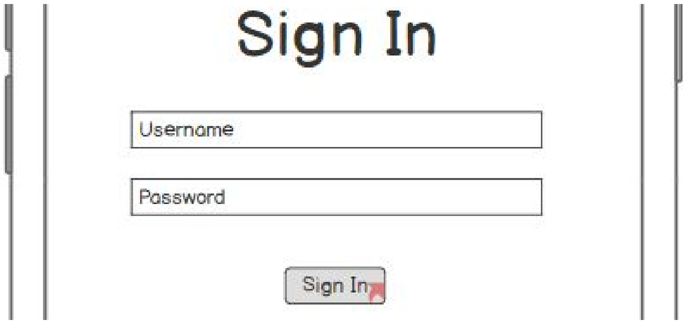
Usability Aspect Reports (UARs)

Reagan #1	**Problem**

Name: Username/password text inputs not nearly large enough for mobile screen

Evidence:

Interface Aspect:



Heuristic: *special consideration for mobile apps

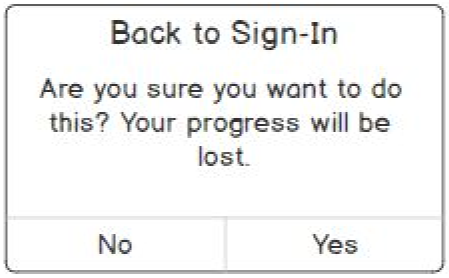
Explanation: The general rule about mobile apps is violated because, in general, it is appropriate to make text/input boxes as large as possible (for usability) while still being aesthetically pleasing. This is because the user must click/select with their finger or even their thumb on the box rather than using a mouse which is much more accurate to click with. These boxes would be very difficult to click on with a finger/thumb on a mobile device.

Severity or Benefit:
Rating: 3
Justification (Frequency, Impact, Persistence):
Frequency: This problem happens a few times on other screens as well, but every single user will experience this problem on the sign-in screen, so it is very frequent.
Impact: I think this problem (which is not uncommon on mobile apps) is typically overcome by the user in about 2-3 seconds if it slows them down at all. It won't be *incredibly* frustrating, but it will be a nagging issue, especially if the user must sign-in to the app every time they use it.
Persistence: This problem will persist often. It isn't possible for the user to fix this problem. So every time they are asked to sign-in, it will be difficult to click on these boxes.
How I weighted the factors: At first, I rated this problem a 2. But after thinking it through some more, I realized that this is a frequent and persistent problem that cannot be avoided by the user and it affects every single user that uses the app. Therefore, I consider this is a major problem that needs to be solved with high priority.

Possible Solution and/or Trade-offs: The easy solution to this problem is, as I mentioned before, make the boxes as large as possible so that it will be easy for the user to click on them, while not making them obnoxiously large or unattractive.

Relationships: N/A

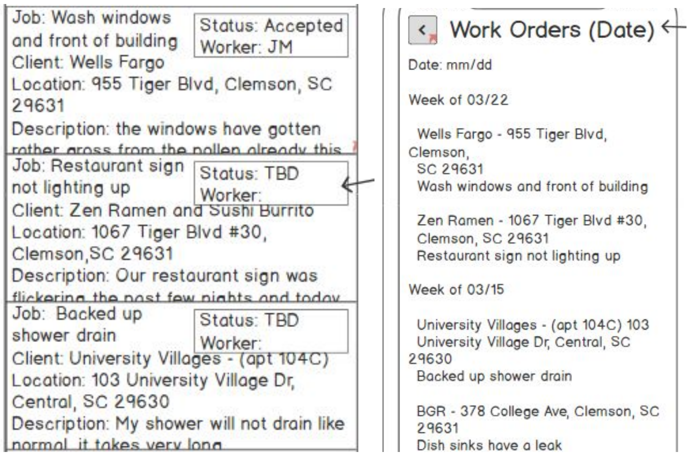
Reagan #2	**Good Aspect**

Name: Good error prevention when confirming if the user wants to “go back and lose progress”
Evidence: **Interface Aspect:** Back to Sign-In Are you sure you want to do this? Your progress will be lost.   **Heuristic:** Error Prevention
Explanation: The heuristic is adhered to because the app is asking to make sure the user knows that if they continue with this action, their progress will be lost. Many users may not have known that the “back” button would erase their progress on this screen, so this is a good method for error prevention.
Severity or Benefit: **Rating:** NA **Justification (Frequency, Impact, Persistence):** This feature will benefit the user by preventing the potential future frustration they would have if they came back from their 2nd trip to the sign-in screen only to find that their progress on the “Create Account” screen was not saved. If not for this error prevention method, the user might unknowingly go back for a trivial reason (maybe just to check something) and then come back to a blank “Create Account” screen and be very upset and have to fill in their info all over again.
Possible Solution and/or Trade-offs: N/A
Relationships: N/A

Reagan #3	**Problem**

Name: Work Orders screen is cluttered and difficult to read

Evidence:
Interface Aspect:



Job: Wash windows and front of building
Status: Accepted
Worker: JM
Client: Wells Fargo
Location: 955 Tiger Blvd, Clemson, SC 29631
Description: the windows have gotten rather gross from the pollen already this

Job: Restaurant sign not lighting up
Status: TBD
Worker:
Client: Zen Ramen and Sushi Burrito
Location: 1067 Tiger Blvd #30, Clemson,SC 29631
Description: Our restaurant sign was flickering the past few nights and today

Job: Backed up shower drain
Status: TBD
Worker:
Client: University Villages - (apt 104C)
Location: 103 University Village Dr, Central, SC 29630
Description: My shower will not drain like normal, it takes very long

Work Orders (Date)

Date: mm/dd

Week of 03/22

Wells Fargo - 955 Tiger Blvd, Clemson, SC 29631
Wash windows and front of building

Zen Ramen - 1067 Tiger Blvd #30, Clemson, SC 29631
Restaurant sign not lighting up

Week of 03/15

University Villages - (apt 104C) 103 University Village Dr, Central, SC 29630
Backed up shower drain

BGR - 378 College Ave, Clemson, SC 29631
Dish sinks have a leak

Heuristic: Aesthetics and Minimalist Design

Explanation: The heuristic is violated because on the first of these 2 screens, there is not nearly enough whitespace, making it very difficult to read and sort out information. The second screen has enough whitespace but is lacking any lines/bars/bolded words/sections to separate any of the information from itself. This will also likely make it very hard for the user to sift through this info.

Severity or Benefit:
Rating: 2
Justification (Frequency, Impact, Persistence):
Frequency: This problem is fairly rare as it only appears for a certain type of user: the maintenance worker. Also, it is only an issue on 2 of 24 of the possible screens for this user.
Impact: This problem does have the potential to waste some time if the worker has difficulty reading the screen and has to squint or really concentrate hard to digest the information.
Persistence: This is a persistent problem as it is not one that the user can work around. The screen is just cluttered and hard to read.
How I weighted the factors: The main reason I rated this a 2 and not a 3 is because of the infrequency of it. It only affects a specific user on a small portion of the time they are using the app. However, it is a fairly bad issue during those times for those users. It would be very nice to fix.

Possible Solution and/or Trade-offs: The solution for this problem is to lessen the amount of information on the screen and make sure that what is there is separated and distinguished clearly. Maybe it would be good to bold important information/headings to distinguish these sections, but on the second screen, definitely add separation bars.

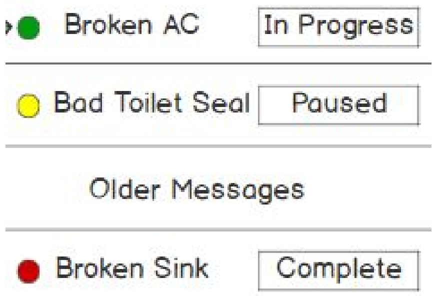
Relationships: N/A

Reagan #4	**Problem**

Name: Colors on Messages screen are not intuitive

Evidence:

Interface Aspect:



🟢 Broken AC In Progress

🟡 Bad Toilet Seal Paused

Older Messages

🔴 Broken Sink Complete

Heuristic: Match between system and real world

Explanation: The heuristic is violated because these colors do not really match what a typical user would expect them to mean. The user would probably expect green to mean the order is complete (fixed), yellow to mean it is in progress, and red to mean the order is paused/on hold because of the connotations associated with these colors. The color green typically means good or operating the way it should, therefore green should mean the order is complete and the problem was fixed. Yellow is simply the transition color between red and green so it should mean "in progress". And the color red generally means something is wrong or stopped, so this should mean "paused".

Severity or Benefit:
Rating: 2
Justification (Frequency, Impact, Persistence):
Frequency: This problem will be common for any user (most) who views their messages regularly.
Impact: This problem will not cause much of a usability issue for users or waste much of their time because the words "in progress," "paused," or "complete" are off to the side of the message as well.
Persistence: This is not an issue that the user can overcome. This is simply a design choice.
How I weighted the factors: The most important factor here is the minimal impact. This problem will affect most users and can't be overcome (though it might be gotten used to), however it doesn't have much of an impact on the typical user. I don't think most users will be *too* annoyed by this so I rated this a minor problem.

Possible Solution and/or Trade-offs: The solution here would be to change the colors as follows: green = complete, yellow = in progress, and red = paused (I would actually recommend using the words "on hold" for red).

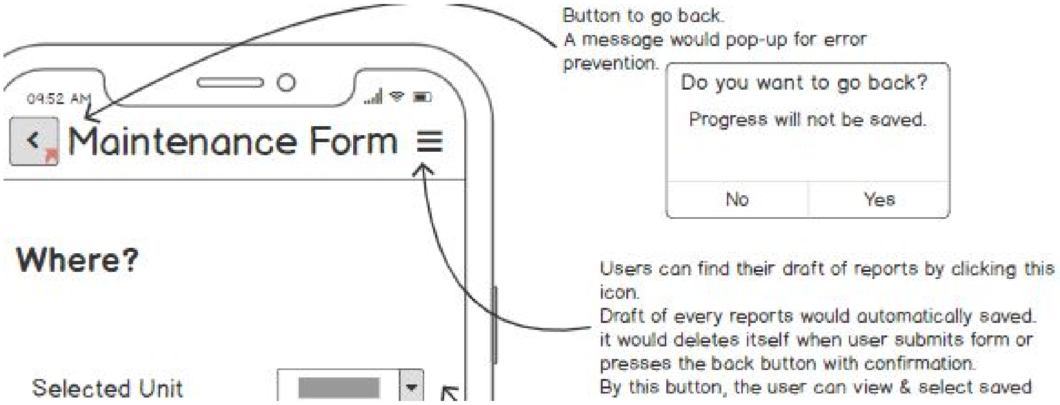
Relationships: N/A

Reagan #5	**Problem**

Name: Conflicting information on whether form progress is saved

Evidence:

Interface Aspect:



Button to go back.
A message would pop-up for error prevention.

Do you want to go back?
Progress will not be saved.

Users can find their draft of reports by clicking this icon.
Draft of every reports would automatically saved.
it would deletes itself when user submits form or presses the back button with confirmation.
By this button, the user can view & select saved

Heuristic: Consistency and Standards

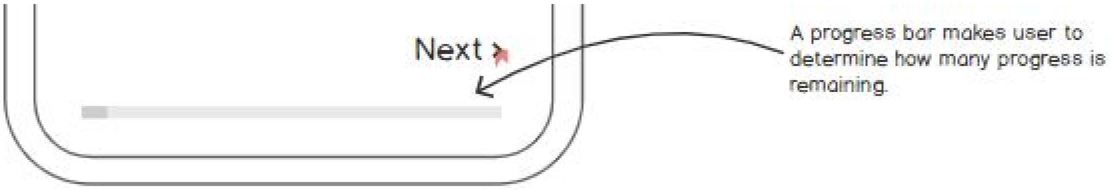
Explanation: The heuristic is violated because there is conflicting information about whether or not the progress on this form is saved. A user could possibly remember that there is a button that will show them their drafts (in the upper right) and when they click the back button (upper left) and are presented with a message saying their progress will not be saved, they would be very confused.

Severity or Benefit:
Rating: 2
Justification (Frequency, Impact, Persistence):
Frequency: This problem might occur to any user who fills out a maintenance form (residents).
Impact: This is not very impactful in that it doesn't hinder the user's abilities to use the app. However, it is very confusing and could cause the user worry over whether or not a draft of this form would be saved.
Persistence: This problem could theoretically be worked around if the user tests that the system saves drafts of the forms. Once the user sees that it does (despite the warning message), they could ignore the warning from then on.
How I weighted the factors: While this is a direct conflict of information within the system and is confusing to the user, I judged it as a minor problem because it does not actually hinder usability of the system and probably wouldn't waste much time. Also, as I mentioned, the user would probably be able to ignore the warning message after knowing that the system saves drafts.

Possible Solution and/or Trade-offs: The easiest solution to this would be to remove the warning message saying that progress will not be saved, because the progress is being saved.

Relationships: N/A

Reagan #6	**Good Aspect**

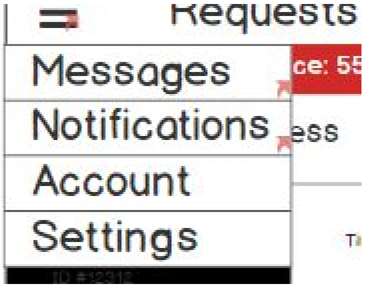
Name: Showing progress bar at the bottom of maintenance request form
Evidence: **Interface Aspect:**    A progress bar makes user to determine how many progress is remaining. **Heuristic:** Visibility of System Status
Explanation: The heuristic is adhered to because it is a great example of showing the user where they are in the completion of the form. Without this progress bar, they have no idea how many pages of the form are left to fill out. This progress bar is a great help to most users who value their time.
Severity or Benefit: **Rating:** NA **Justification (Frequency, Impact, Persistence):** This feature will help every user who fills out a maintenance form by letting them know how much further they need to go to complete this form. Features like this one that improve visibility of system status often minimize user frustration and/or lengthen their patience if the user knows there's only x pages left of the form.
Possible Solution and/or Trade-offs: N/A
Relationships: N/A

Reagan #7	**Problem**

Name: Not clear or easy to change account email or phone number

Evidence:

Interface Aspect:



Heuristic: User Control and Freedom

Explanation: The heuristic is violated because there are no screens shown beyond either the “Account” or “Settings” buttons here, and so it is not obvious how the user is to change their account information such as their contact info, or even if they are able to (to me, this is a form of “undoing an action”). In reality, I’m sure this group had intended for this feature to be included in their system. However, it just isn’t clear in these wireframes if that is the case.

Severity or Benefit:
Rating: 3
Justification (Frequency, Impact, Persistence):
<u>Frequency:</u> It will not be often that a user needs to change their contact information so this won’t be a very frequent problem.
<u>Impact:</u> This would, however, have a tremendous impact, essentially rendering the app useless if the user’s phone number changed and they were not able to change it easily or at all.
<u>Persistence:</u> If, in fact, it is possible for the user to change this information, then they will likely remember how to change it the next time they might need to. So this won’t be a particularly persistent problem.
<u>How I weighted the factors:</u> I rated this as a major problem because of the impact that it could potentially have on the system. As I said, if it is not possible to change this information, then the user would have to create an entirely new account with a new username and password or potentially call IT support just to change their contact info (terribly inconvenient).

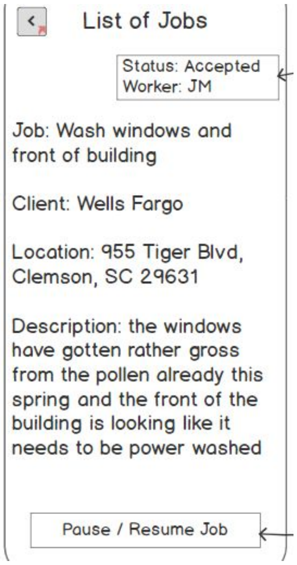
Possible Solution and/or Trade-offs: The solution to this is to have a quick and easy option to “Update Account Info” on either the “Account” or “Settings” screens.

Relationships: This is related to Problem #8 as they both prevent the user from seeing an action that may be available to them.

Reagan #8	**Problem**

Name: No clear way for worker to accept a job

Evidence:
Interface Aspect:



List of Jobs

Status: Accepted
Worker: JM

Job: Wash windows and front of building

Client: Wells Fargo

Location: 955 Tiger Blvd, Clemson, SC 29631

Description: the windows have gotten rather gross from the pollen already this spring and the front of the building is looking like it needs to be power washed

Heuristic: Recognition rather than recall

Explanation: The heuristic is violated because the system is not showing every action available to the user (accepting a job). This is similar to the previous problem in that there clearly must be an option to accept a job (or the system wouldn't work), but it is not at all clear how the maintenance worker can do this.

Severity or Benefit:
Rating: 3
Justification (Frequency, Impact, Persistence):
Frequency: This problem will be more frequent for "worker" users as they will likely need to accept new jobs every day or every other day.
Impact: This would have a rather large impact on workers if it is really difficult to accept a job.
Persistence: This would be a persistent problem because even if the user can find a workaround, they will still have to perform this workaround every time they need to accept a job (which is frequent).
How I weighted the factors: The main factor here is the frequency. This is a major problem because it will inevitably affect every "worker" user nearly every day. Therefore, it essentially making the app frustrating to use for an entire subsection of users.

Possible Solution and/or Trade-offs: The solution to this problem is to have a button on the Job screen (shown above) to "Accept Job" if it has not already been accepted.

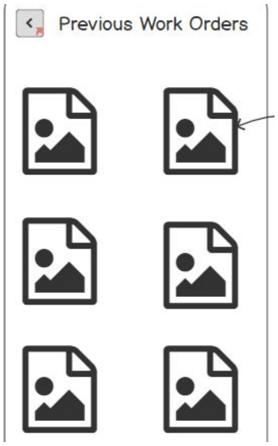
Relationships: This is related to Problem #7 as it is also does not show the user all their options.

Reagan #9	**Problem**

Name: No option to sort previous work orders (or see how they are sorted by default)

Evidence:

Interface Aspect:



Heuristic: Flexibility and efficiency of use

Explanation: The heuristic is violated because advanced users (who may want to see these previous work orders sorted a certain way) have no option to see these work orders how they would like to. There isn't even an easy way to see how they are sorted by default (no dates or information underneath each picture).

Severity or Benefit:
Rating: 2
Justification (Frequency, Impact, Persistence):
Frequency: This is a fairly infrequent problem as it only affects "worker" users and they probably wouldn't need to view previous work orders on a daily basis.
Impact: This doesn't have a vast impact on the usability of the app. It would just be nice to have.
Persistence: This would not be a very persistent problem as there are probably dates and other information contained on the work order itself. So the user could just click on the picture to see when/where it was from.
How I weighted the factors: I saw this as a minor problem/nice-to-have because it is a minor part of the app and is for a small, specific set of users (advanced maintenance workers). It is not particularly frequent, impactful, or persistent. So this is a minor problem.

Possible Solution and/or Trade-offs: The solution to this problem would be to have a date underneath each picture and a sort button at the top of the page with at least 2 options ("Sort by most recent" and "Sort by least recent").

Relationships: N/A

Conclusion

Most of the issues that I found with the system were due to features being much too small for a mobile device interface. In my opinion, essentially every text box, dropdown, and checkbox are much too small for a thumb/finger to click on. However, this doesn't not really fall under one of Nielsen's ten usability heuristics, so I mentioned it in only one of my UARs so I could focus on other design aspects relating to these heuristics. The other problems that I found were a few cluttered screens with insufficient lines/bars to separate sections from one another, some confusing colors for the job statuses on the Messages screen, some conflicting information given by a warning message within the maintenance form, and the lack of easy-to-find and easy-to-use ways for workers to accept jobs, sort work orders, and for all users to change account/contact information.

A summary of a few suggestions I have to improve the design, functionality, and overall usability of the system is as follows: increase the size of all text boxes/dropdowns/checkboxes throughout the system, reduce clutter/increase segmentation of work order screens, change colors of message statuses to be more intuitive, remove warning message from maintenance form, and provide easy and user-friendly options to change account/contact information, for workers to accept jobs, and for workers to sort previous work orders. Implementing all of these design improvements would enhance the system greatly as a whole and make for a much more user-friendly and free environment. It would also be ideal to be able to implement these features while also maintaining the good features of the system that I observed, which were good error prevention and good visibility of system status.

Reflection

In this design critique, I found seven problems with Group 9's wireframes with four of them being minor and three of them being major. I believe that I found most of the issues with the system design and I believe that the issues that I found were the most pertinent and in most need of fixing. There are certainly more issues to be found (as there almost always are), but not crucial ones. As I went through the wireframes to find these issues, once I found one, I completed the UAR in its final form. So rather than go through finding all the problems and writing rough UARs to go back and finalize them later, I would spend as much time as I needed to write the finalized UAR initially. I would say that each UAR took about 30 minutes or so to complete from start to finish.

Nielsen's heuristics were quite helpful in finding these problems. However, I found that some problems did not fall under any of them specifically (at least not clearly). For instance, the first problem I found--namely, that the text boxes/dropdowns/checkboxes were too small for fingers to click on--was more of a general design principle for mobile devices. So while Nielsen's heuristics are a helpful guide for finding design issues, they do not always catch everything. Additionally, I think that the UARs are a very useful way to describe usability problems. I especially think that the "Justification" section of the UAR is most helpful in determining the severity of any given problem by weighing the "Frequency", "Impact", and "Persistence" against one another. I think this is the best way to do this.

If I were to do this assignment again, I think the only thing that I would do differently would be to find the problems with the design before looking for any of the "good aspects" which were easier to find. I initially looked for both simultaneously, which I think made me less likely to find either in actuality. Searching for just the problems initially would have made for a quicker and more efficient and effective analysis.