HEURISTIC EVALUATION

DEBIT MEMO TRACKER

DASHBOARD-1

NO ISSUE TO SELECT   MINOR

VISIBILITY OF SYSTEM STATUS

If the top table is empty, and the user is new—which is more than likely—it could be confusing what they should do to move forward. The bottom table says select an issue, but there is no issue to select. One possible fix for this is to use a message similar in placement to the bottom panel's message, informing the user that they'll have an issue present in their list once one is assigned to them.

DRAWING A BLANK   MINOR

VISIBILITY OF SYSTEM STATUS

When a user uses the search bar and they search for something that returns no results, it will be important to display a message (perhaps in the middle of the table, perhaps as a toast message, or something else) informing them that their search returned no result. Otherwise they may think the application isn't working correctly.

WHAT (ISSUES) AM I LOOKING AT?   MINOR

VISIBILITY OF SYSTEM STATUS

When the Mine toggle is selected, its relatively clear which issues are showing. It could be made crystal clear if there was a small explanation

of what "Mine" meant. This explanation could be in a smaller, lighter font above the table, or (maybe and) as a tooltip on the toggle.
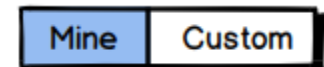
We face a more difficult problem with the other half of the toggle: "Custom." I chose this wording because I couldn't think of way to succinctly describe the concept of not just your issues, but whatever set of issues you specify. Maybe this concept would be made clearer with an established metaphor.

ISSUE DETAILS   COSMETIC

MATCH BETWEEN SYSTEM AND REAL WORLD

Looking at the current system, one large communal email group, the phrase "Issue Details" isn't used, so that poses a small risk.

W NUMBERS   MAJOR

MATCH BETWEEN SYSTEM AND REAL WORLD

Two user groups will use this application, QC and Marketing. QC uses a PO (Purchase Order) number—they refer to it as "PO" in conversation. Marketing cares much less for PO numbers and instead uses something called a W number. As of now, there is no W number in the table.

ONCE PULLING ISSUES YOU CAN'T STOP UNTIL YOU'RE DONE — MINOR
USER CONTROL AND FREEDOM

When a user specifies which set of issues they would like to see, and they submit that, it will start pulling down those issues and it won't stop (unless you shut down the program, I guess). This could cause an issue if it takes an excessively long time—granted, there is a progress spinner pane that covers everything and makes it clear that the program hasn't frozen—but you know, people are impatient.

TRYING TO ADD TO AN ISSUE WHEN NONE IS SELECTED — MAJOR
ERROR PREVENTION

In the wireframe right now, I'm showing the "+" FAB even though the issue has not been selected yet. This might lead someone to try to add something to an issue that they think is selected. To overcome this, I'll make sure the FAB is hidden when there is no issue selected.

ALL CLICKS — COSMETIC (IN A FUNCTIONAL WAY)
FLEXIBILITY AND EFFICIENCY OF USE

As the application stands, almost everything is just accessible by mouse click. The exception to this is being able to tab between things, but the scratch on that is that the tab order isn't super consistent or predictable. Accelerators (made known to the user by an underlined character where there is text or via a tooltip where there is not) would HELP efficiency.

ISSUE NUMBER COLUMN — MINOR
AESTHETIC AND MINIMALIST DESIGN

I'm unsure if the user will care about the issue number that I have there in the first column. They don't have issue numbers in their current system, they just refer to issues by their item number.

TOOLTIPLESS — MAJOR
HELP AND DOCUMENTATION

A good way to show tooltips on a wireframe hadn't occurred to me. They're a very interactive feature. However, I feel like they'll play a key role in giving the user more information about how to use the application. Maybe, I can show all the tooltips on their own wireframe.

DASHBOARD-2

NOT ENOUGH INFO — MAJOR
VISIBILITY OF SYSTEM STATUS / RECOGNITION RATHER THAN RECALL

Once a user clicks on an issue, all the details for that issue pop up. This section was inspired by the issue tracker on GitHub. In the wireframe there's some but not enough detail about each

event. It still needs to be brief, but it will be important to show key info here, otherwise the user might end up having to *always* click the DETAILS button to see what they need to.

In many of the company's applications, to accomplish something, the user right clicks on a row. I've yet to include that functionality in the upper table. All the actions are currently accomplished with the "+" floating action button

As it stands, once a user does an action (like writing a comment) there's no getting rid of it. Some actions are "reversible" in the sense that they can be done but in the other direction. One example of such a reversible action is assigning the issue to someone: you can assign the issue back to yourself afterwards. My inclinations for not allowing this (yet) feel technical but maybe there's a way around it—and maybe it's more needful than I've considered.

To create this application, I've been going by Google's Material Design Guidelines. In those guidelines, the table columns have not separator. I'm afraid this could cause clarity problems if any column's content is ever too long for the column and runs right into the next one. It will wrap, of course, but having it so close with no separator could be confusing.

All the buttons that appear out of the Floating Action Button are the same size as the original. This might lead some to think that the plus button functions like the rest of the buttons. This isn't the case, however, that button will just open and close the speed dial menu.

The Material Design Guidelines describe the top app bar thusly, "The top app bar displays information and actions relating to the current screen." I don't think I've done an excellent job of following that. So far, I've mostly used the top bar for the title.

Also, I'm under the impression that the ellipsis icon should only appear if there is something that it could show. I think I'll either put some actions into that top menu or get rid of the ellipsis menu where it's not needed.

REASSIGNED ISSUES LOST INTO THE VOID   MAJOR
RECOGNITION RATHER THAN RECALL

As it stands, once you reassign an issue to someone else, there's no easy way to keep track of it. You can look for issues under their name, but maybe they already reassigned it. You can look it up by P.O.#, but you'll need to have the P.O.# for that; same goes for item number. You can find it, but there is no easy way to follow up with issues you've recently been involved in.

ICON MEANING   MAJOR
### RECOGNITION RATHER THAN RECALL

Right now, there's nothing to tell you what will happen when you click one of the floating action buttons. It would be helpful, I think, if right when you moused over a popup appeared with the task name, something short.

QUICK ACTIONS WITH THE FABS   MINOR
FLEXIBILITY AND EFFICIENCY OF USE

I'm not sure exactly how to use accelerators with purely visual icons, as there's no way to make it clear how to use them. I guess I could use a tooltip, but that kind of clashes with my proposal for the Icon Meaning problem, above. Perhaps I could have all the shortcuts appear on a cheat sheet when the user presses "?". StackOverflow, GitHub, and even Gmail do this, so there's precedent. We'll have to find a way, however, to let the user know that they need to press "?" to get a look at all the shortcuts.
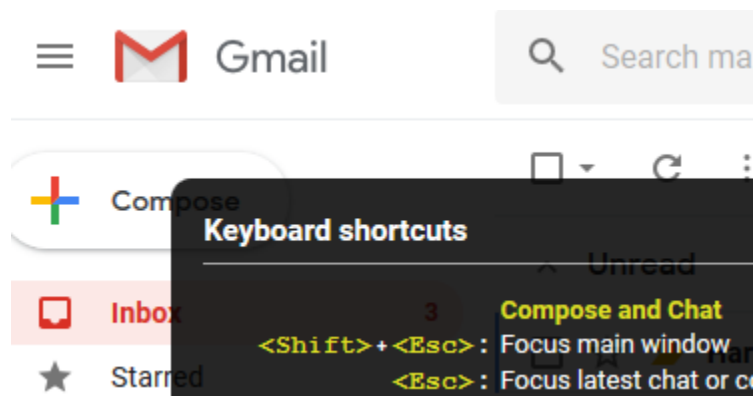
ICON FONT   COSMETIC
AESTHETIC AND MINIMALIST DESIGN

In my application, I happen to use the exact icon font that is available on Balsamiq: FontAwesome. However, I'm putting into practice lots and lots of Material Design guidelines. This makes me wonder if I would be better off using a set of Material Icons instead.

TOOLTIPLESS   MAJOR
HELP AND DOCUMENTATION

A good way to show tooltips on a wireframe hadn't occurred to me. They're a very interactive feature. However, I feel like they'll play a key role in giving the user more information about how to use the application. Maybe, I can show all the tooltips on their own wireframe.

DEBIT-MEMO-1

VALIDATION FEEDBACK — MINOR

VISIBILITY OF SYSTEM STATUS

Right now, some fields are restricted to only allow the right kind of input. If a user tries to put in a letter where a number is expected, it won't allow that. The problem is that it doesn't really tell you anything, it just says, "No," by deleting the offending character. Maybe the user needs some sort of feedback right then to explain.

ALL DONE DIALOG — MINOR

VISIBILITY OF SYSTEM STATUS

When the user submits on Debit-Memo-1, we should show them a dialog after they've submitted that lets them know that they've saved things and that they'll now return to the Dashboard.

DEBIT FREIGHT — MINOR

MATCH BETWEEN SYSTEM AND REAL WORLD

I chose this wording because that seemed to match the idea in but few words. The issue is that

this isn't a common phrase for them, they usually use quite a few more words to describe this. Maybe I can make this unmistakably clear by adding a tooltip using more words.

FULL MEASURES ONLY — MAJOR

USER CONTROL AND FREEDOM

Right now, a user can only create a whole debit memo, they can't draft one leaving parts empty. I feel like this will be a major pinch point.

SUBMIT TO WHO? — MINOR

CONSISTENCY AND STANDARDS

Looking at the bottom of the Debit-Memo-1 screen, we see that the button says "SUBMIT." Thinking about the whole, I wonder if that will

trip some people up, you know, because they're not ready to submit it to the higher ups, you know, they just want to save it.

COPY ADDRESS — MINOR

RECOGNITION RATHER THAN RECALL

Many times, I suspect, a vendor will have the same billing and shipping address. It would support user's memory load by allowing the user an easy way to copy over the whole address to the *Ship To* section.

TAB ORDER AND ACCELERATORS — MINOR

FLEXIBILITY AND EFFICIENCY OF USE

On this screen there's a lot of data to be input. It would be most idea if the user is allowed to do the whole screen without taking their fingers off the keyboard. Adding accelerators to the to the options and ensuring the tab order is correct will ensure that.

TOOLTIPLESS	MAJOR

HELP AND DOCUMENTATION

A good way to show tooltips on a wireframe hadn't occurred to me. They're a very interactive feature. However, I feel like they'll play a key role in giving the user more information about how to use the application. Maybe, I can show all the tooltips on their own wireframe.

SOME INSTRUCTION	MINOR

HELP AND DOCUMENTATION

There is little instruction on this page. Maybe it would be good to have at least a little instruction placed there at the top to supplement the instruction scattered throughout (inside the fields and tables).

DASHBOARD-3

NOT ENOUGH INFO	MAJOR

VISIBILITY OF SYSTEM STATUS / RECOGNITION RATHER THAN RECALL

Once a user clicks on an issue, all the details for that issue pop up. This section was inspired by the issue tracker on GitHub. In the wireframe there's some but not enough detail about each event. It still needs to be brief, but it will be important to show key info here, otherwise the user might end up having to *always* click the DETAILS button to see what they need to.

DEBIT MEMO CREATION WORDING	MINOR

MATCH BETWEEN SYSTEM AND REAL WORLD

Right now, debit memos are created by hand, and once they're created, they also happen to be on a piece of paper. It might be important to adjust the wording accordingly to account for the difference between created, approved, and printed.

Jane Doe **created a Debit Memo** on Apr 02, 2019

REMOVE DEBIT MEMO	MINOR

USER CONTROL AND FREEDOM

Once a user creates a debit memo, there's no way to get rid of it. I can't help but wonder if they'll want to do that. (But it makes it more complex from a technical standpoint!! 😅) If there is the option to delete it, I think it would be best to put that functionality in the Debit Memo pane, perhaps as a trash icon in the app bar. I'll have them type the word DELETE to get rid of it, just to be sure.

QUICK ACTIONS WITH THE FABS	MINOR

FLEXIBILITY AND EFFICIENCY OF USE

I'm not sure exactly how to use accelerators with purely visual icons, as there's no way to make it clear how to use them. I guess I could use a tooltip, but that kind of clashes with my proposal for the Icon Meaning problem, above. Perhaps I could have all the shortcuts appear on a cheat sheet when the user presses "?". StackOverflow, GitHub, and even Gmail do this, so there's precedent.

TOOLTIPLESS	MAJOR

HELP AND DOCUMENTATION

A good way to show tooltips on a wireframe hadn't occurred to me. They're a very interactive feature. However, I feel like they'll play a key role in giving the user more information about how to use the application. Maybe, I can show all the tooltips on their own wireframe.

DEBIT-MEMO-2

READ-ONLY CONTROLS	MAJOR

VISIBILITY OF SYSTEM STATUS

When the user is on this screen, they're only viewing the debit memo. We should make sure that anything that looks like they can edit it comes with an affordance that makes it clear that they can't—or, if possible, make it look like they *can't* edit it in the first place.

Also, like was mentioned in the book. When the field is read-only, it would be extra informative to give each one a tooltip showing how the user can make it editable.

It looks like they can change the toggle, but they cannot.

FIELD LABELS	MAJOR

VISIBILITY OF SYSTEM STATUS

One thing that I need to change in the wireframes is the kind of input field. I recently figured out how to put the Material style of field, and there's even an option to show the floating field label.

EDITABLE DEBIT MEMO	MAJOR

MATCH BETWEEN SYSTEM AND REAL WORLD

If a user ever wants to edit a debit memo, they'd have to create a whole new one. Instead, why don't we allow them to edit an existing one and save it as a whole new one. We could do this in this Debit-Memo-2 Wireframe maybe by putting a pencil icon up in the app bar.

PRINT OR EMAIL ONLY ONCE APPROVED	MAJOR

MATCH BETWEEN SYSTEM AND REAL WORLD

We probably shouldn't allow the debit memo to be emailed or printed until it's approved, right? We wouldn't want unapproved debit memos getting sent.

AUTHORITY TO AUDIT · MAJOR

The person looking at this screen, may be the person approving or disapproving the debit memo. We should allow them to change virtually anything, even things the computer has put there. I feel like this will be important in the case that the computer is wrong, and we need to account for it.

PRINT PREVIEW · MAJOR

Surely, even before the debit memo is approved, we'll want to let the user see what it would look like in paper form. Maybe we can accomplish this by one or maybe both of the following. First, we could put a DRAFT or UNAPPROVED watermark on the generated PDF. As another form of precaution, I'm almost certain that one can restrict the printability of a PDF. Using both is the safest bet.

ACCELERATORS · MINOR

I'm a little hesitant to put accelerators on the Disapprove and Approve buttons—also, according to the style guide those should be in all caps—but maybe I will anyways.

Also, worth mentioning is the ability to activate any of the actions in the top bar by accelerator. Maybe something that would be useful would be to imitate Microsoft word's style of accelerator, by putting the letter that one must push over each control whenever the user holds down the Alt key.

SOME INSTRUCTION · MINOR

Surely, we should give the user some instruction about what their task is on this page. Some well-picked words should go a long way.

NEXT STEPS · MINOR

Once the user finishes this screen it might be good to offer some suggestion of some next steps.