Chat Threads product Requirements, Feb. 13, 2024

https://moodysanalytics.atlassian.net/browse/GIGE-2139

There is a list of Product requirements that I uncovered, while thinking through the Conversation History:

CONTEXT:

Conversation History and Chat threads is a complex feature. As mentioned below, it contains several functions. Previous Jira was #1898. Can we please take a look together and align?

1. **For clarification of nomenclature:** "Conversation History" is the new side drawer. "Chat Thread" is one single chat thread; one item in the side drawer.

2. **An alternative to the 2 side drawers, is utilizing one side drawer, with 2 tabs on the top:** Entitlements, and Conversation History. But in this case, you never see both selections at the same time, but you toggle between the 2. Product, please indicate if you prefer this direction.

3. **We are implementing the "Selected" label and "Hide unselected" toggle**, on the Entitlements Side Drawer in this Sprint. Product and Dev, please confirm.

Conversation History and chat threads' features:

Need Product to confirm this is the desired functionality, and Dev to confirm this functionality is feasible. Bold are individual features listed:

1. **Collapsible Conversation History side drawer:** User can see the Thread, and the Entitlements corresponding with that thread, at the same time.

2. **Automated, immediate Entitlements selection is triggered, when user selects a Chat Thread:** When a user selects a thread, they see the Entitlements selection happening in front of their eyes. Which is a good reminder of what Ents are selected, and that this change is happening. Product and Dev, please confirm.

3. **Chat thread names are created automatically by the model.** Same way ChatGPT works. Please see "ChatGPT_Chat_Threads.mov" video for how that works. Product and Dev, please confirm.

Copyrights © Yelena Dobric 2024. Do not download, copy or distribute.

4. **Implement time buckets:**

   • Chat GPT has: “Today”, “Previous 30 Days”. After that, in ChatGPT it’s “2023”.

   • For our use case, UX suggests: “Previous 60 Days”, “Previous 90 Days”, “Past 6 months” and “Past 12 months”. Then as we get closer to 2025, we can create a folder called “2024”. Product and Dev, please confirm.

5. **Long names fade out towards the edge:** In ChatGPT, long thread names “fade out”. Please see “ChatGPT_Chat_Threads.mov” video for how that works. Alternative is truncating. Product and Dev, please clarify preference.

6. **Chat-thread based behavior, on hover and click:** When user hovers and clicks, names are faded out even more, to reveal row-based actions. Hover shows “More” tooltip, 3 dots, and the “Archive” icon. Please see “ChatGPT_Chat_Threads.mov” video for how that works. Product and Dev, please confirm if you want/can code this.

7. **“Rename”, “Delete” and “Archive” are a chat-thread based functions**. Please see “ChatGPT_Chat_Threads.mov” video for how that works. There is also “Share” - are we ready to incorporate sharing? Product and Dev, please confirm if you want/can code this.

8. **Recognition of existing Threads triggers relevant warning message to the user, and ability to go to that Thread:** Interaction and mutual dependency between Entitlements, and Threads. Scenario: user clicks “New chat”, but then manually selects a combination of Entitlements, that they already have in a Thread. They get a warning message “You already have a chat thread with this combination of Entitlements. Would you like to go to it now? CTAs: “Yes, take me there” and “No, I’’ll start fresh”.” Product and Dev, please confirm.

9. **A new scenario triggers an existing warning message:** “Please select Entitlements to chat with.” User is in a thread, and deselects all Entitlements. Product and Dev, please confirm.

CREATING:

1. **Automatically:** user clicks “New chat”. They select one Entitlement, or more. And starts chatting. The new chat thread is created automatically. Just like ChatGPT does it.

Copyrights © Yelena Dobric 2024. Do not download, copy or distribute.

PROS: User does not have to remember to save a thread.

CONS: Use has to delete a thread, if they don't want it.

OR

2. **Manually:** user clicks "New chat". They select one Entitlement, or more. And starts chatting. But they have to explicitly click somewhere to "Save this chat thread", for example.
   PROS: In this case, we eliminate the need to explicitly have to "Delete" threads they don't want.
   CONS: But they have to remember to "Save".
   SAVING:
   The same concept applies to how not just how a Thread starts. But how it "lives":

3. **Automatically:** user is almost always in a thread. And whatever edits they make, while in a thread, the Skill just saves that LAST SET OF SETTINGS. They can select and deselect Entitlements, they can "Clear chat", whatever they do. We just save that.
   OR

4. **Manually:** user has to remember to "Save edits". Or we give them messages and popouts.
   UX recommendation is automated behavior. Yes, users may need to get used to it. But "getting used to it" only happens once. It's a pattern we call "learned behavior" in UX. After that one time, user will likely be annoyed with messages and popouts.

Copyrights © Yelena Dobric 2024. Do not download, copy or distribute.