

Heuristic Evaluation of Jar

1. Problem

Jar is an application that allows roommates to assign tasks to each other and hold everyone accountable monetarily.

2. Violations Found

1. H2-3. User control and freedom [Severity: 4] [Found by: A, B, C]

Task 1: The interface of creating/assigning a task doesn't allow the user to cancel creating the task. If the user wants to cancel midway, he has to keep clicking the back button until he reaches back to the home screen.

Fix: Add a "cancel" button to each screen of the sequence.

2. H2-10. Help and documentation [Severity: 1] [Found by: A, C]

Task 1: It is not clear to the user that how the task is going to be assigned: does it assign the task to the user himself, does it randomly assign to someone in the group, or is it based on everyone's current tasks?

Fix: Add one-line description beneath the "assign task" button.

3. H2-2. Match between system and the real world [Severity: 2] [Found by: A, B, C]

Task 1: It is confusing at first that there's no profile picture showing for the current user's tasks. I understand that it is good to differentiate the current user's tasks from other users', but the empty circle looks like it is for some user who didn't upload his profile picture.

Fix: I suggest using different color or fonts, or filling the circle with added text "YOU" to specialize the current user's tasks. This may seem unnecessary since there is another panel displaying "My Tasks", but this subtle change can highlight, or provide a direct comparison between the user's tasks and those of the other users.

4. H2-1. Visibility of system status [Severity: 0] [Found by: A]

Task 1: The user who creates the task is able to see whom the task has been assigned to, but if the task is not assigned to the user himself, the task owner needs to be notified right away as well.

Fix: Add notification to inform user if he has a newly assigned task.

5. H2-7. Flexibility and efficiency of use [Severity: 4] [Found by: A, B]

Task 1: When the user reaches the end of the complete task flow before submitting and notices incorrect input, he has to go all the way back to the corresponding page to edit and then walk through the other pages until he gets to the overview page again.

Fix: Make the fields shown on the overview page clickable and navigate users to the corresponding page to update.

6. H2-7. Flexibility and efficiency of use [Severity: 3] [Found by: A, C]

Task 1: The user needs to go through 4 screens to input all the required fields when creating a task. It would be troublesome for the user to input information for the same common tasks each time. This is especially annoying because the user is likely to forget what tasks he/she came up with after going through the entire process.

Fix: Save a list of top 5 common tasks from history (or by user's preferences) to accelerate the process of creating/assigning a task. Also allow multiple inputs in task settings interface.

7. H2-4. Consistency and standards [Severity: 2] [Found by: A, C]

Task 2: In the "transfer request sent" dialogue, there are a cross button on upper right corner and an "ok" button, both for acknowledgement purpose. If the user wants to cancel the transfer ("request" at the moment), he/she has to click "OK" anyway and click "Cancel Transfer" in the next interface. If the confirmation box only contains one button, then it can be designed as a pop-up message.

Fix: Remove the redundant cross button or adding an "Undo" button to the confirmation box.

8. H2-5. Error prevention [Severity: 3] [Found by: A, C]

Task 2: After clicking on a person in the transfer task screen, the transfer request is immediately sent. User cannot cancel the request if the user clicks the wrong person by accident. Although there is a "cancel transfer" button shown after the request has been sent, it is better to present the user a confirmation option before he commits to action, because cancelling request and cancelling transfer should be two different states.

Fix: Change the request sent dialogue to a confirmation dialogue with "cancel" and "send" buttons before sending the request.

9. H2-1. Visibility of system status [Severity: 4] [Found by: A]

Task 2: Although the user can tell that the transfer request has been accepted when the task is shown with another user's profile picture, there is no indication of whether the transfer request he sent has been declined or the other user hasn't responded yet.

Fix: Add a notification to inform the user when the transfer request is accepted/declined, and ability to see status of transferred task within app (accepted, declined, pending).

10. H2-2. Match between system and the real world [Severity: 2] [Found by: A]

Task 3: All users in the group need to approve to use the money in the jar.

Fix: Add a dialogue/notification to send to all users in the group for approval before using the money.

11. H2-1. Visibility of system status [Severity: 2] [Found by: A]

Task 3: The user is not able to keep track of how money was spent through jar.

Fix: Add history page to the complex task to help user keep track of recent transactions through jar.

12. H2-10. Help and documentation [Severity: 1] [Found by: A]

General: There's no documentation to help the user understand what will happen if he doesn't complete his assigned task, how much money does he need to pay for not completing a task in time, where does the money in the jar come from?

Fix: A "help" button in the menu page to provide necessary information.

13. H2-6. Recognition rather than recall [Severity: 0] [Found by: A]

General: The user needs open the app to check if he has an upcoming deadline of a task. User might forget to keep track of his tasks and miss deadlines.

Fix: It would be good to send reminders to the user when the deadline is near.

14. H2-2. Match between system and the real world [Severity: 1] [Found by: B]

Upon opening the app, users are directly taken to "all tasks." I imagine that users are less interested in viewing other people's tasks and are more interested in the status of their own tasks and how much time they have left, to make sure they do not lose money to the jar.

Fix: Make the first screen start on "My Tasks" instead, then the user can have the option to navigate

to view everyone else's tasks.

15. H2-7. Flexibility and efficiency of use [Severity: 1] [Found by: B]

After marking "Task Completed," a popup appears which then reroutes us back to the details page with all of the buttons deactivated. Since there's nothing to do and nothing left to see on this page, rerouting back here is extremely inefficient.

Fix: Rather than going back to the details, it should take you straight back to the tasks page.

16. H2-7. Flexibility and efficiency of use [Severity: 3] [Found by: B]

After marking "Task Completed," the user cannot unmark the task as completed from within the details page that he/she is taken back to. If one wants to unmark a task because they accidentally pressed completed, he/she has to route back out to the tasks view and uncheck the box next to the task.

Fix: Rather than deactivating the buttons, change them so that one of the buttons stays active and allows users to reverse that action.

17. H2-2. Match between system and the real world [Severity: 2] [Found by: B]

Navigating to your own task's detail page via the "All Tasks" page and then clicking the back arrow at the top left brings the users to the "My Tasks" page when they were expecting to be routed back to "All Tasks."

Fix: Make sure that navigating backwards goes to the most previously visited screen to avoid confusion.

18. H2-4. Consistency and Standards [Severity: 4] [Found by: B, C]

After transferring a task, the "Transfer Task" button goes gray, which is the same action for when the buttons become deactivated after completing a task.

Fix: Rather than making the button change to gray, have it change to another color that isn't consistent with the idea of an inactive button, like red or yellow.

19. H2-2. Match between system and the real world [Severity: 3] [Found by: B, C]

When creating a task, the user is drawn from the top bottom to fill out information on each page, and then has to go back up to the top of the screen to move on to the next view, which is confusing and inefficient.

Fix: Rather than making users zigzag up and down the screen, put a next button at the bottom under the information input so that the task flow moves in one direction consistently, as the user would expect.

20. H2-2. Match between system and the real world [Severity: 1] [Found by: B]

In the view that shows the jar, the jar is empty, but the number floating above it indicates that it contains money.

Fix: Add a graphic that shows that the jar is actually filling up when there is money in it to be more consistent with the physical idea of an actual jar filling up with money.

21. H2-3. User control and freedom [Severity: 4] [Found by: B, C]

After creating a task, there is no way to edit or delete the task that you created. This could be an issue if someone accidentally creates a task without realizing that it was already completed, or realizes that they created it with incorrect information.

Fix: When clicking on a task, allow users to edit or delete, as long as all other users agree to the edit or

deletion.

22. H2-1: Visibility of Status [Severity: 1] [Found by: C]

In the simple task “Create and Assign a Task”, the captions “Task Name”, “Deadline”, and “Duration” are somehow redundant in a sense that the instructions on the interface are self-explanatory. However it is not obvious to see the status of current stage.

Fix: Changing those captions to “Task Settings” might be more helpful in a sense that the user can see the main status here is setting up the task.

23. H2-1: Visibility of Status [Severity: 2] [Found by: C]

After creating the task and clicking “Assign Task” button, there will be a pop-up window showing task assigned to “You”. The point is if the user clicks “X” instead of “OK” button, the pop-up window will disappear and the user would not directly know whether the task is assigned or cancelled; maybe he/she can confirm that by checking the task list, but it is inconvenient.

Fix: Display a status message after clicking either “OK” or “X”.

24. H2-1: Visibility of Status [Severity: 2] [Found by: C]

The “Add Notes...” is really helpful because it allows users to add reminders, or warnings for the task. However in the case of multiple transfers, how to deal with different notes added by each user is not specified in the prototype.

Fix: Add a notes editing pop-up window to complete the function.

25. H2-1: Visibility of Status [Severity: 3] [Found by: C]

The “Complete Task” button is a little bit confusing here since the task is already transferred to someone else.

Fix: I suggest disabling the “Complete Task” button when the task is transferred, which means in order to click “Complete Task”, the user has to cancel the transfer beforehand. This feels like a more formal definition.

26. H2-3: User Control & Freedom [Severity: 3] [Found by: C]

When a transfer request is received, the user will face a pop-up window with only two options: “Accept” or “Deny” before he/she can do anything. This is annoying because it is forcing the user to make a decision right away, and it increases the chance of error.

Fix: I recommend adding a “Later” button, and all received requests should be collected in a list for later review, with appropriate notifications.

27. H2-5: Error Prevention [Severity: 2] [Found by: C]

When creating a task, the “Deadline” and “Duration” panels allow users swipe left/right to move backward/forward in the process. However they both do not contain buttons to save/update the status.

Fix: Add save status buttons.

28. H2-8: Aesthetic & Minimalist Design [Severity: 1] [Found by: C]

The interface contains five colors: red, blue, black, grey, and white. While I strongly appreciate the simple, minimalist interface, this may seem a little bit monotonous for some users.

Fix: I suggest keeping the minimal style, and add a few more colors. For example, use orange to indicate upcoming tasks (less urgent than red tasks even closer to deadlines), or green fill color for some tab/panels. Use color code to indicate different urgent levels (tasks, requests, etc).

29. H2-10: Error Prevention [Severity: 0] [Found by: C]

While using money in JAR, if the user is close to a Apple Pay reader and clicks “Use” by mistake, the app cannot undo the transaction.

Fix: I suggest adding a confirmation box, as well as some up-limit for one time transaction, similar to credit card protection.

3. Summary of Violations

Category	# Viol. (sev 0)	# Viol. (sev 1)	# Viol. (sev 2)	# Viol. (sev 3)	# Viol. (sev 4)	# Viol. (total)
[H2-1: Visibility of Status]	1	1	3	1	1	7
[H2-2: Match Sys & World]		2	3	1		6
[H2-3: User Control]				1	2	3
[H2-4: Consistency]			1		1	2
[H2-5: Error Prevention]			1	1		2
[H2-6: Recognition not Recall]	1					1
[H2-7: Efficiency of Use]		1		2	1	4
[H2-8: Minimalist Design]		1				1
[H2-9: Help Users with Errors]						0
[H2-10: Documentation]	1	2				3
Total Violations by Severity	3	7	8	6	5	29

4. Evaluation Statistics

Severity / Evaluator	Evaluator A	Evaluator B	Evaluator C
sev. 0	66.6%	0%	33.3%
sev. 1	28.6%	28.6%	42.9%
sev. 2	50%	25%	75%
sev. 3	33.3%	33.3%	83.3%
sev. 4	60%	80%	60%
total (sev. 3 & 4)	5	6	8
total (all severity levels)	13	11	17

5. Summary Recommendations

Jar does a good job of maintaining a simple and clean interface while presenting all of the necessary information. One source of confusion had to do with the functionality of different aspects in the “Tasks” view and the results of actions on this page, mainly with regards to how one’s own tasks were differentiated from others’ tasks. The flow of creating tasks doesn’t allow any user freedom, since there is no way to cancel or easily edit a task while creating, or edit and delete a task after creation. This functionality is essential if one main purpose of the application is to easily keep track of assigned tasks. Another similar issue is about error prevention, like when buttons appear inactive or users are not able to keep track of the statuses of certain tasks. As a summary the application is well-designed. Due to limitations of test cases and prototype implementation, some of the functionalities are left out, and it will help clarify and streamline tasks when fully implemented. Advanced usage such as creating multiple tasks at the same time, getting multiple tasks transferred from/to multiple people has not been tested. These are all usability/convenience issues worth considering in the next phase of interface design.

Appendix

Severity Ratings

- 0 - Not a usability problem, but a feature to consider adding
- 1 - cosmetic problem
- 2 - minor usability problem
- 3 - major usability problem; important to fix
- 4 - usability catastrophe; imperative to fix

Heuristics

[H2-1: Visibility of System Status]

- keep users informed about what is going on

[H2-2: Match Between System & Real World]

- speak the users' language
- follow real world conventions

[H2-3: User Control & Freedom]

- "exits" for mistaken choices, undo, redo
- don't force down fixed paths

[H2-4: Consistency & Standards]

[H2-5: Error Prevention]

[H2-6: Recognition Rather Than Recall]

- make objects, actions, options, & directions visible or easily retrievable

[H2-7: Flexibility & Efficiency of Use]

- accelerators for experts (e.g., gestures, kb shortcuts)
- allow users to tailor frequent actions (e.g., macros)

[H2-8: Aesthetic & Minimalist Design]

- no irrelevant information in dialogues

[H2-9: Help Users Recognize, Diagnose, & Recover from Errors]

- error messages in plain language
- precisely indicate the problem
- constructively suggest a solution

[H2-10: Help & Documentation]

- easy to search
- focused on the user's task
- list concrete steps to carry out
- not too large