Team R Project: Stage 4

08.23.2020

— CPSC 481

Sebastien Wong - 30040657

Danny Tran - 30027536

Noah Bensler - 30020684

Shamim Khalili - 30022743

Term: Fall 2020

TA: Philmo Gu

Team R Portfolio

https://sebastienwong.github.io/cpsc481teamr/portfolio/

Team R Repository

https://github.com/sebastienwong/cpsc481teamr

Updated Project Idea

The project we have chosen is a grocery shopping mobile app. With the app, users will be able to easily add items to their shopping list by scanning the barcode of the items or searching by name. Users can also remove items from the list and automatically tally the price of each item in order to keep track of budget. This will allow users to easily keep track of exactly what they purchase and the total cost. As the app is used over the months, it will keep track of the users spending habits, allowing them to easily track their grocery budget. The app will also provide features such as recommending coupons for frequently purchased items, reorganizing the list to group similar items, and allow users to manually add and remove items that don't have barcodes. This app will be useful for anyone who buys groceries regularly, but especially for people who buy large amounts of groceries at once, people shopping on a tight budget, and people with poor memory such as the elderly.

List of updated tasks

TASKS IMPLEMENTED VERTICALLY

- User will add/remove items in their shopping list.
- User can track their purchases and change their budgets.
- User can search for and apply coupons to their list.
- User can create new shopping list templates.

REMAINING HORIZONTAL TASKS

- User can organize their list by aisle/store.
- User can track health and nutrients of their groceries.

Heuristic Evaluation Process

The Heuristic Evaluation process was completed using the heuristic evaluation form. This process involved completing ten steps which each evaluated a different aspect of the prototype. The prototype was evaluated one step at a time, finding all the instances where each rule was either applied or violated. Each time this occurred, a brief description was written, detailing the application or violation. The evaluation was done by Noah and Danny.

The results that were obtained from this process were consistent between the two independent evaluations. Both found that the prototype lacked functions to prevent errors from occuring or to easily recover from errors once they had occurred. They also found that many of the buttons lacked functionality. Overall, the evaluations were positive and found that nearly all the rules were applied consistently throughout the prototype.

Heuristic Evaluation Review/Findings (the debriefing part)

The process of reviewing was rather straightforward, and Sebastien and Shamim acted as the reviewers for heuristic evaluation. After reviewing and sorting the issues separately, we found that we shared many of the same observations and lots of the issues were graded with similar severity.

Our findings revealed that the most severe problems were issues where functionality was expected but not delivered. For instance, check boxes not being able to be unchecked, or non functional back buttons. We agreed these would be the most severe, as they are needed to keep the flow of the prototype functional and understandable. We agreed that the least severe issues we're issues regarding documentation or error messaging. While these issues are important, for a user, this information isn't super important to them, they just want an app that is intuitive to use and will work. And in the middle ground of severity, we agreed that issues regarding speed and convenience belonged here. These issues do not break the functionality of the app, but they can help improve usability and flow, and should not be unaccounted for.

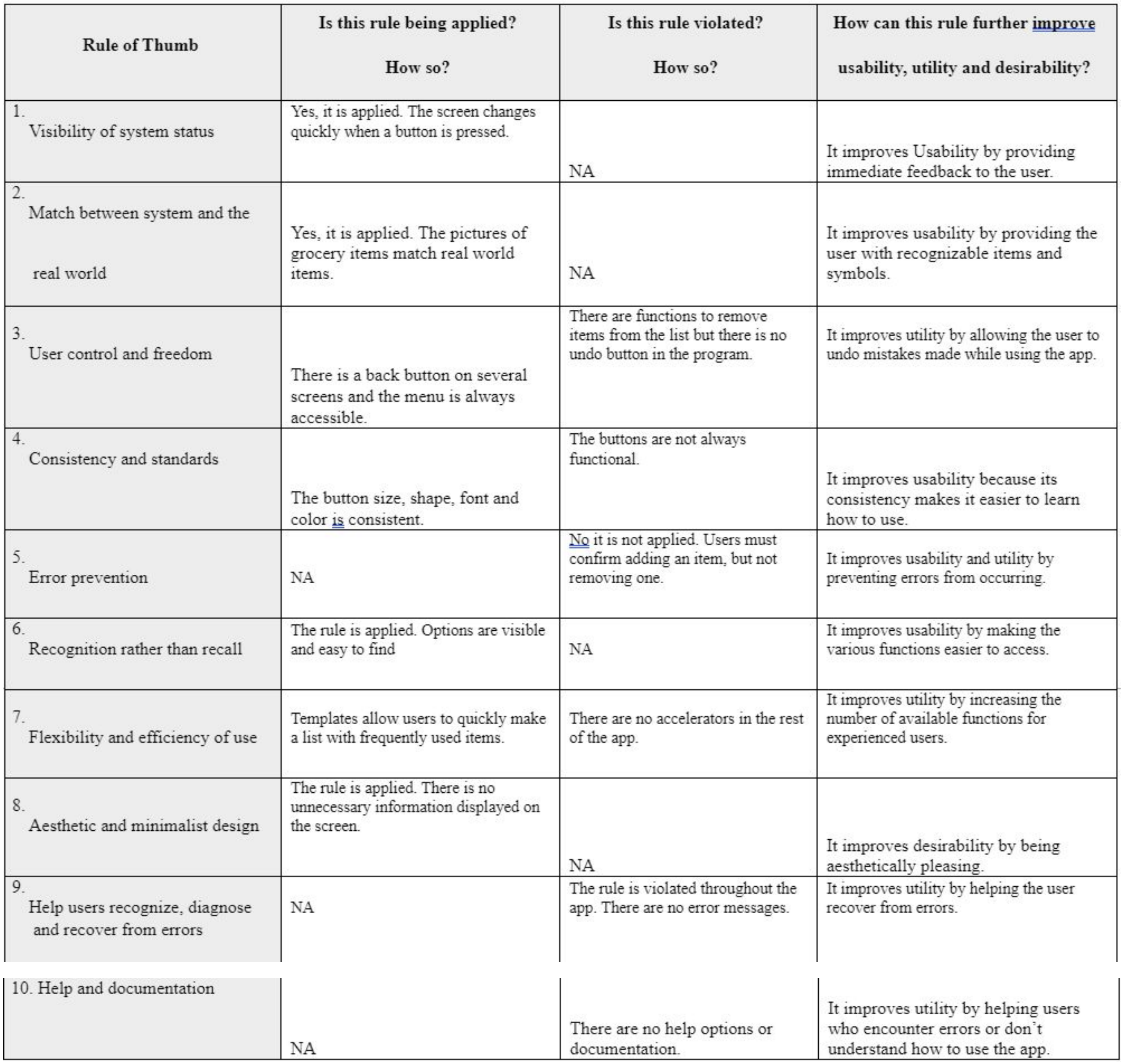
Reflections

During the Heuristic Evaluation Process we learned that user testing was a must when it comes to developing an application, because there are so many things that you don't consider when building it. For example, when we made all the buttons for our app, we didn't think about making them look unclickable because we thought that it wouldn't be a big deal for a prototype, but when we tested it, it was very jarring to use. From that we learned that all the general things (like buttons) that a user expects to work when using an app, must also work when they are using a high fidelity prototype.

Given more time, we would have liked to make the design more appealing and minimalistic, but we didn’t have enough time to do so.

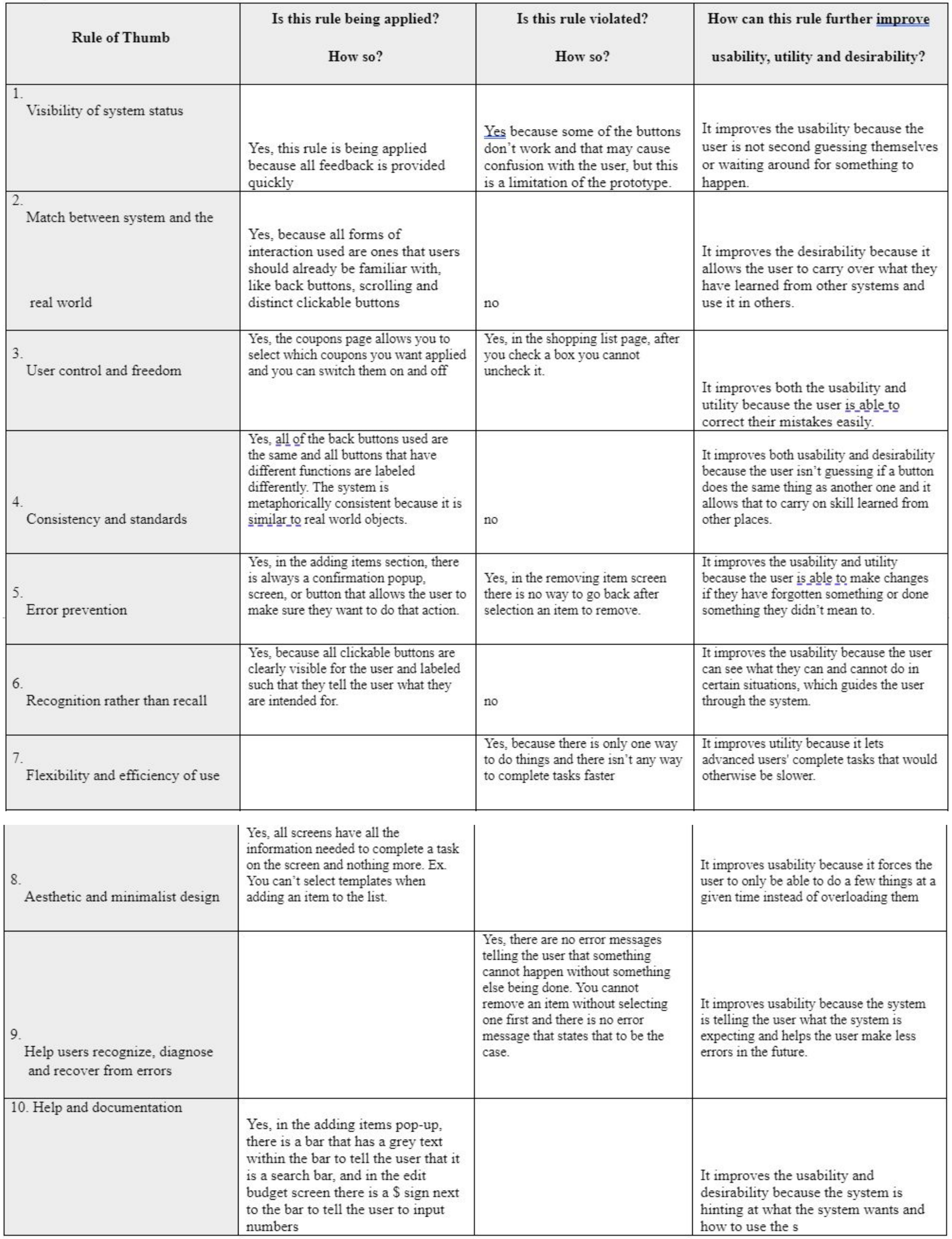
If we were to do this process again, we would change our approach to our prototype to be general usability focused first compared to main functionality focused because that would save us more time in the long run if we make errors. For our Heuristic Evaluation next time, we would like a third party to do this kind of evaluation in order to eliminate any developer bias during the evaluation as we feel that we could learn a lot more if we had more new eyes.

Appendix



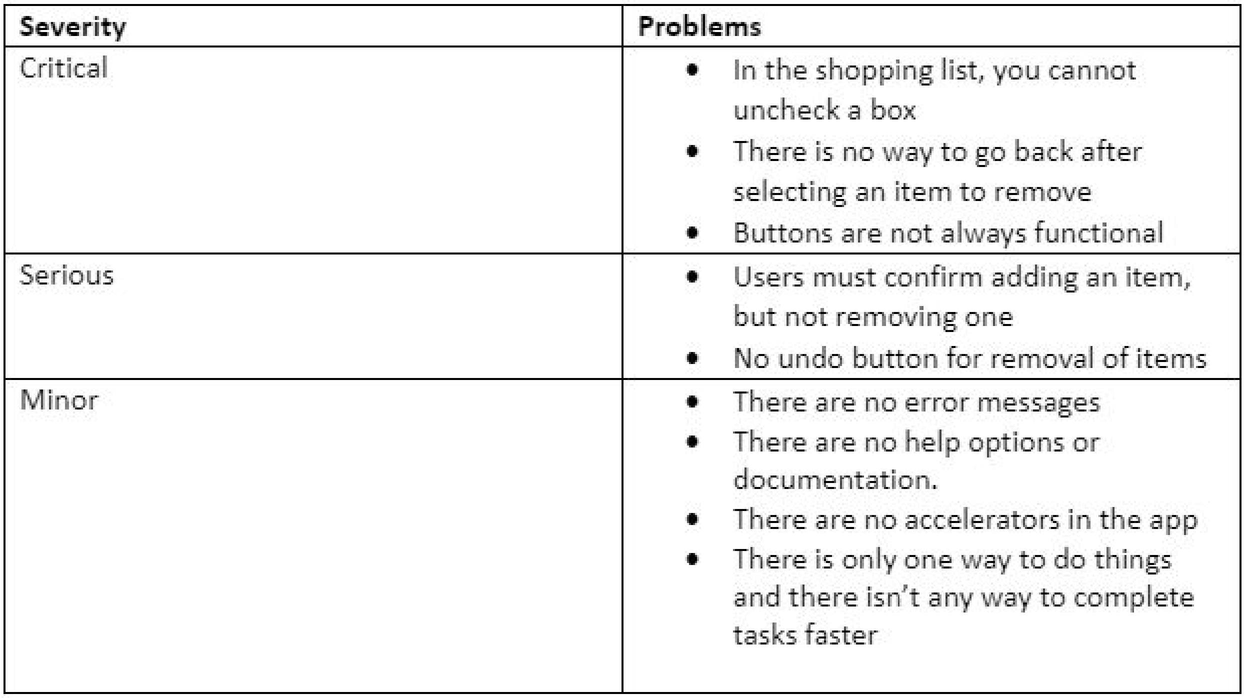
Rule of Thumb	Is this rule being applied? How so?	Is this rule violated? How so?	How can this rule further improve usability, utility and desirability?
1. Visibility of system status	Yes, it is applied. The screen changes quickly when a button is pressed.	NA	It improves Usability by providing immediate feedback to the user.
2. Match between system and the real world	Yes, it is applied. The pictures of grocery items match real world items.	NA	It improves usability by providing the user with recognizable items and symbols.
3. User control and freedom	There is a back button on several screens and the menu is always accessible.	There are functions to remove items from the list but there is no undo button in the program.	It improves utility by allowing the user to undo mistakes made while using the app.
4. Consistency and standards	The button size, shape, font and color is consistent.	The buttons are not always functional.	It improves usability because its consistency makes it easier to learn how to use.
5. Error prevention	NA	No it is not applied. Users must confirm adding an item, but not removing one.	It improves usability and utility by preventing errors from occurring.
6. Recognition rather than recall	The rule is applied. Options are visible and easy to find	NA	It improves usability by making the various functions easier to access.
7. Flexibility and efficiency of use	Templates allow users to quickly make a list with frequently used items.	There are no accelerators in the rest of the app.	It improves utility by increasing the number of available functions for experienced users.
8. Aesthetic and minimalist design	The rule is applied. There is no unnecessary information displayed on the screen.	NA	It improves desirability by being aesthetically pleasing.
9. Help users recognize, diagnose and recover from errors	NA	The rule is violated throughout the app. There are no error messages.	It improves utility by helping the user recover from errors.
10. Help and documentation	NA	There are no help options or documentation.	It improves utility by helping users who encounter errors or don't understand how to use the app.

Figure A. Team R Heuristic Evaluation done by Noah



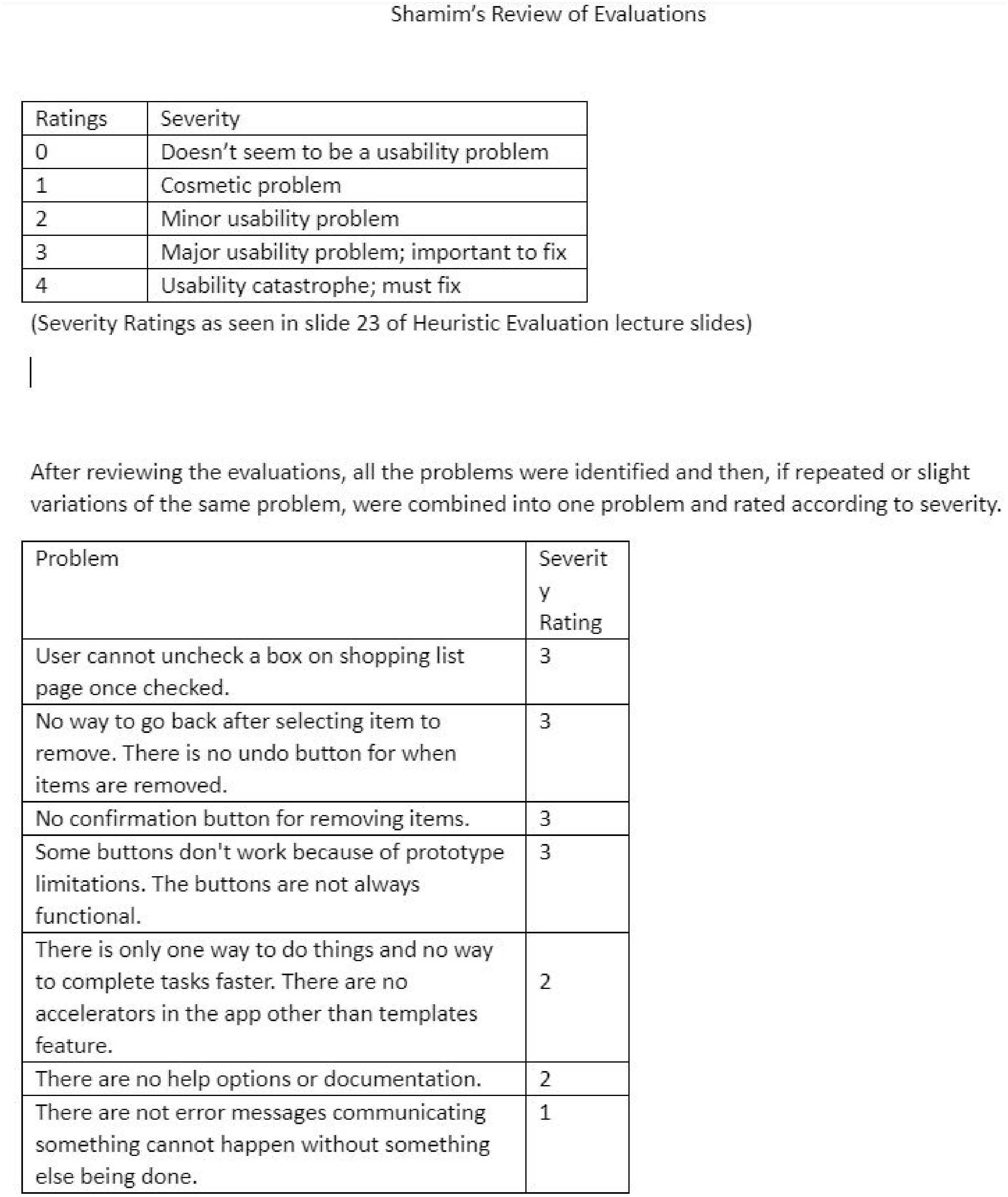
Rule of Thumb	Is this rule being applied? How so?	Is this rule violated? How so?	How can this rule further improve usability, utility and desirability?
1. Visibility of system status	Yes, this rule is being applied because all feedback is provided quickly	Yes because some of the buttons don't work and that may cause confusion with the user, but this is a limitation of the prototype.	It improves the usability because the user is not second guessing themselves or waiting around for something to happen.
2. Match between system and the real world	Yes, because all forms of interaction used are ones that users should already be familiar with, like back buttons, scrolling and distinct clickable buttons	no	It improves the desirability because it allows the user to carry over what they have learned from other systems and use it in others.
3. User control and freedom	Yes, the coupons page allows you to select which coupons you want applied and you can switch them on and off	Yes, in the shopping list page, after you check a box you cannot uncheck it.	It improves both the usability and utility because the user is able to correct their mistakes easily.
4. Consistency and standards	Yes, all of the back buttons used are the same and all buttons that have different functions are labeled differently. The system is metaphorically consistent because it is similar to real world objects.	no	It improves both usability and desirability because the user isn't guessing if a button does the same thing as another one and it allows that to carry on skill learned from other places.
5. Error prevention	Yes, in the adding items section, there is always a confirmation popup, screen, or button that allows the user to make sure they want to do that action.	Yes, in the removing item screen there is no way to go back after selection an item to remove.	It improves the usability and utility because the user is able to make changes if they have forgotten something or done something they didn't mean to.
6. Recognition rather than recall	Yes, because all clickable buttons are clearly visible for the user and labeled such that they tell the user what they are intended for.	no	It improves the usability because the user can see what they can and cannot do in certain situations, which guides the user through the system.
7. Flexibility and efficiency of use		Yes, because there is only one way to do things and there isn't any way to complete tasks faster	It improves utility because it lets advanced users' complete tasks that would otherwise be slower.
8. Aesthetic and minimalist design	Yes, all screens have all the information needed to complete a task on the screen and nothing more. Ex. You can't select templates when adding an item to the list.		It improves usability because it forces the user to only be able to do a few things at a given time instead of overloading them
9. Help users recognize, diagnose and recover from errors		Yes, there are no error messages telling the user that something cannot happen without something else being done. You cannot remove an item without selecting one first and there is no error message that states that to be the case.	It improves usability because the system is telling the user what the system is expecting and helps the user make less errors in the future.
10. Help and documentation	Yes, in the adding items pop-up, there is a bar that has a grey text within the bar to tell the user that it is a search bar, and in the edit budget screen there is a $ sign next to the bar to tell the user to input numbers		It improves the usability and desirability because the system is hinting at what the system wants and how to use the s

Figure B. Team R Heuristic Evaluation done by Danny



Severity	Problems
Critical	In the shopping list, you cannot uncheck a box There is no way to go back after selecting an item to remove Buttons are not always functional
Serious	Users must confirm adding an item, but not removing one No undo button for removal of items
Minor	There are no error messages There are no help options or documentation. There are no accelerators in the app There is only one way to do things and there isn't any way to complete tasks faster

Figure C. Team R Heuristic Review done by Sebastien



Shamim's Review of Evaluations

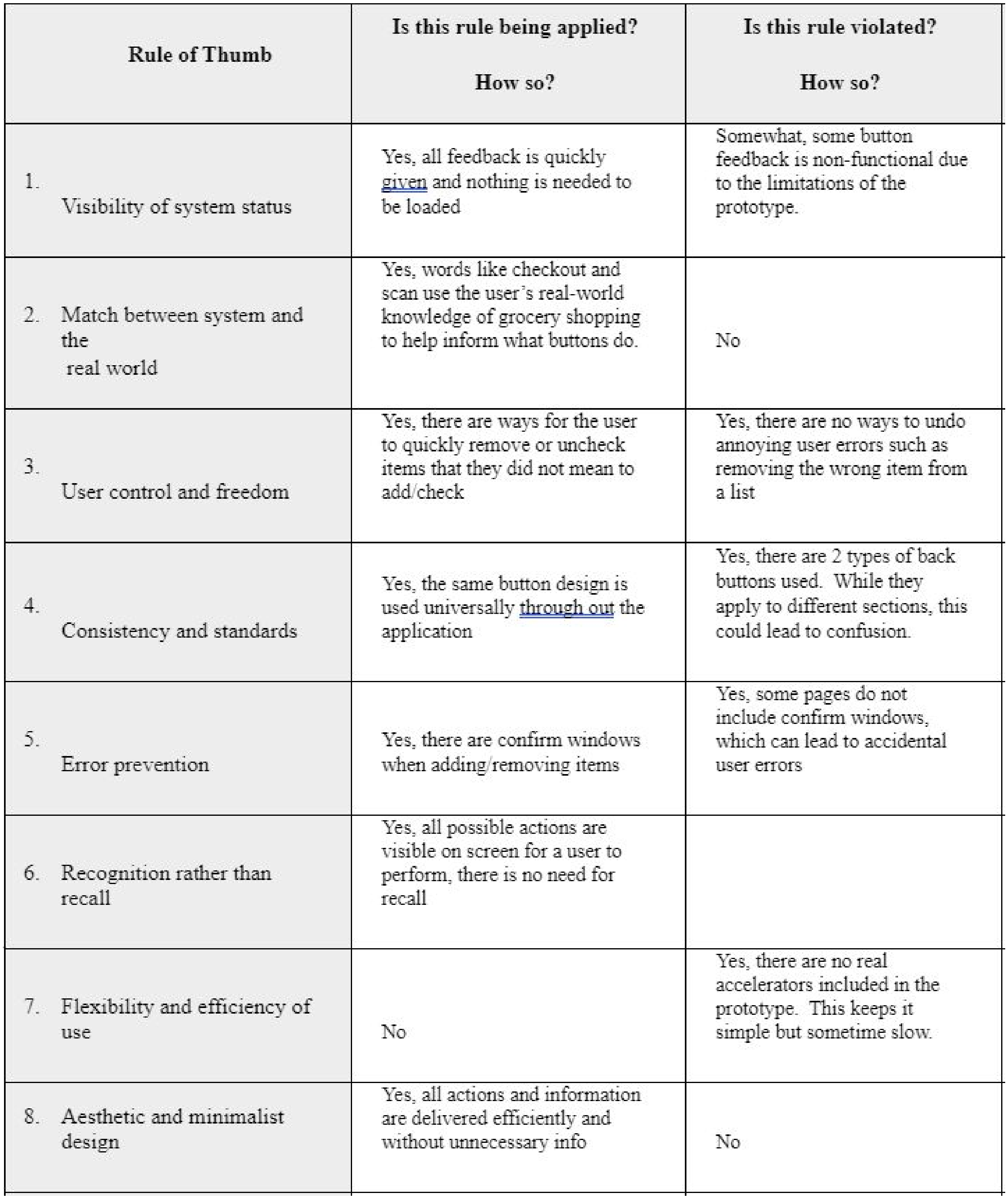
Ratings	Severity
0	Doesn't seem to be a usability problem
1	Cosmetic problem
2	Minor usability problem
3	Major usability problem; important to fix
4	Usability catastrophe; must fix

(Severity Ratings as seen in slide 23 of Heuristic Evaluation lecture slides)

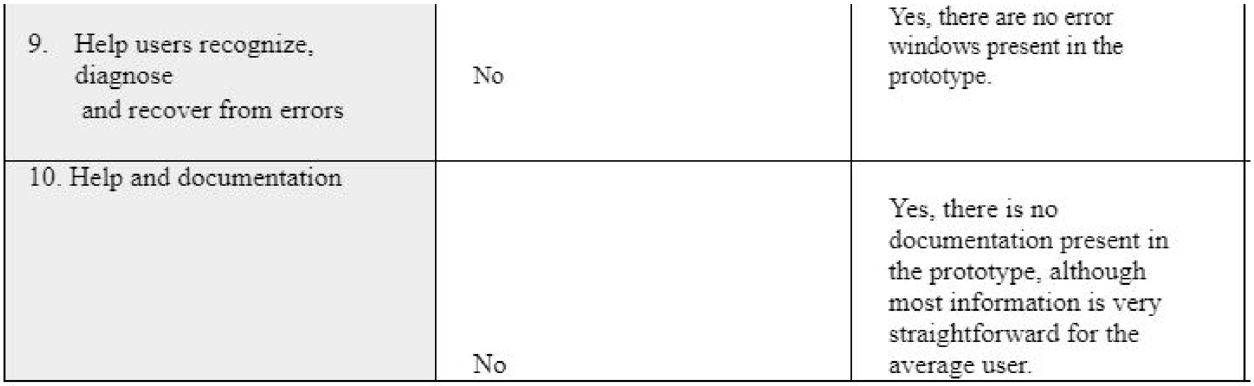
After reviewing the evaluations, all the problems were identified and then, if repeated or slight variations of the same problem, were combined into one problem and rated according to severity.

Problem	Severity Rating
User cannot uncheck a box on shopping list page once checked.	3
No way to go back after selecting item to remove. There is no undo button for when items are removed.	3
No confirmation button for removing items.	3
Some buttons don't work because of prototype limitations. The buttons are not always functional.	3
There is only one way to do things and no way to complete tasks faster. There are no accelerators in the app other than templates feature.	2
There are no help options or documentation.	2
There are not error messages communicating something cannot happen without something else being done.	1

Figure D. Team R Heuristic Review done by Shamim



Rule of Thumb	Is this rule being applied? How so?	Is this rule violated? How so?
1. Visibility of system status	Yes, all feedback is quickly given and nothing is needed to be loaded	Somewhat, some button feedback is non-functional due to the limitations of the prototype.
2. Match between system and the real world	Yes, words like checkout and scan use the user's real-world knowledge of grocery shopping to help inform what buttons do.	No
3. User control and freedom	Yes, there are ways for the user to quickly remove or uncheck items that they did not mean to add/check	Yes, there are no ways to undo annoying user errors such as removing the wrong item from a list
4. Consistency and standards	Yes, the same button design is used universally through out the application	Yes, there are 2 types of back buttons used. While they apply to different sections, this could lead to confusion.
5. Error prevention	Yes, there are confirm windows when adding/removing items	Yes, some pages do not include confirm windows, which can lead to accidental user errors
6. Recognition rather than recall	Yes, all possible actions are visible on screen for a user to perform, there is no need for recall	
7. Flexibility and efficiency of use	No	Yes, there are no real accelerators included in the prototype. This keeps it simple but sometime slow.
8. Aesthetic and minimalist design	Yes, all actions and information are delivered efficiently and without unnecessary info	No



9. Help users recognize, diagnose and recover from errors	No	Yes, there are no error windows present in the prototype.
10. Help and documentation	No	Yes, there is no documentation present in the prototype, although most information is very straightforward for the average user.

Figure E. Team R Final Heuristic Evaluation done by everyone