Milestone 4

Group 7

Zeel Khokhariya

(Khokharz@myumanitoba.ca)

Sangmin Lee

(Lees3430@myumanitoba.ca)

Tinotenda Mpofu

(Mpofut1@myumanitoba.ca)

Zach Wolfe

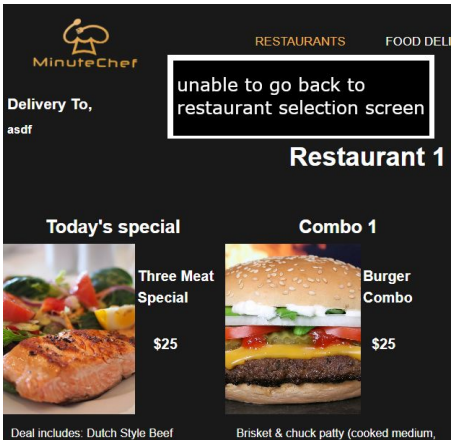
(Wolfez@myumanitoba.ca)

Method

We involved four participants (our group members) in our heuristic evaluation method. Each participant individually used all parts of the system, at each step consulting the list of Nielsen's heuristics and taking note of any violations. After each participant was satisfied with their list of violations, we scheduled a Zoom meeting. During our meeting, we discussed every discovered violation, and made sure that each of them really did violate the heuristics the participant claimed they did. We also discussed what kind of resolution we would recommend for these problems. Next, we grouped similar and identical violations together, and sorted the complete list in order of descending severity. Finally, we assigned each group member a list of violations to more thoroughly document in the section below.

Core issues

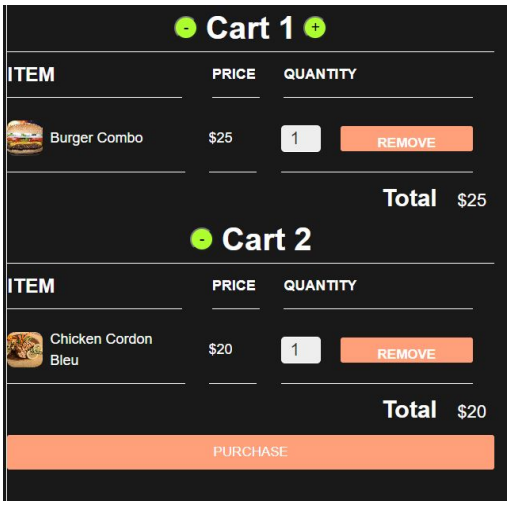
Lack of User Control and Freedom



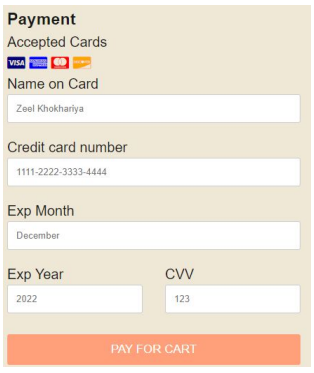
After advancing to any given step of the food ordering process, there is no way to go back to the previous step. For example, if the user wanted to change their order after proceeding to the payment page, there is no way for them to do that. Similarly, after selecting a restaurant, there is no way to go back to the restaurant selection screen. The only option the user has in such a situation is to click the MinuteChef logo, which takes them back to the address entry page and deletes all of their progress: their address, their choice of restaurant, any items added to their cart, and any payment information added. This is a critical usability problem, and as the title suggests, is a violation of the “user control and freedom” heuristic.

To fix this problem, we recommend adding a back button to the interface. This would help both users who unintentionally advance to the next step, and users who intentionally advance, but later realize that they need to correct something from an earlier step. On the whole, this would greatly enhance the usability of the system.

Lack of Error Prevention



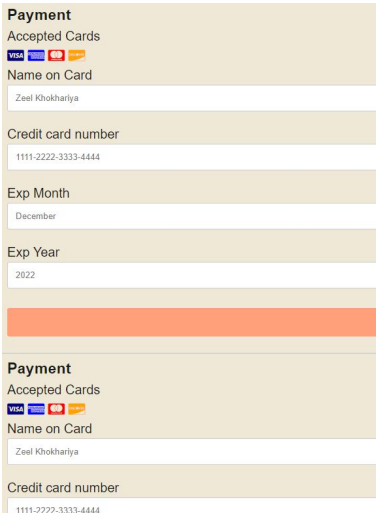
When the user clicks the “PURCHASE” button or later, the “PAY FOR CART” button, they are not asked to confirm this action unless the order is invalid (for example, if there are one or more empty carts). This has the potential to lead to both slips and mistakes. Users may accidentally click a button before they are ready to perform the given action, despite having an accurate mental model of how the system works. This would be a slip. Alternatively, the user may create and fill multiple carts and expect the “PURCHASE” button to only facilitate the purchase of one of them at a time, instead of what it actually does, all of them. This would be a mistake.



To fix this problem, we recommend displaying a confirmation dialog box every time the user clicks the “PURCHASE” or “PAY FOR CART” buttons. This would reduce slips by demanding the user’s attention, and could be designed to reduce mistakes as well, by including a hint as to

how the purchase button works as part of its message. It should also be noted that if there were a back button, this problem would not be as bad, because advancing to the purchase screen prematurely would be far less destructive to the user's progress.

All Payment Information Entered On One Screen

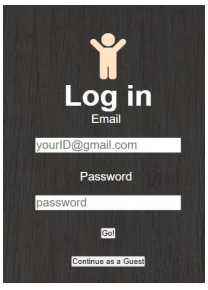


When the user creates and fills multiple carts, they need to pay for each of them on the same physical screen. This violates the heuristic "flexibility and efficiency of use", for multiple reasons. Only one payer can enter their payment information at a time, so the process could be very slow for a large group. Also, unlike with the delivery address, there is no way to save payment information. This reduces efficiency for frequent users. Another consequence of each payer entering their information on the same screen is reduced flexibility. For example, if each payer could pay on their own devices, they are much more likely to have payment information already stored somewhere. This may also make payers feel more secure about entering their information.

To solve this problem, we recommend two improvements. First, add an option to save payment information. Second, for each cart, offer several payment options. First, manual entry should remain possible. Second, the user can enter an email address to send the bill to. Finally, the user can choose a previously-saved payer.

Two Different Login Screens Look The Same

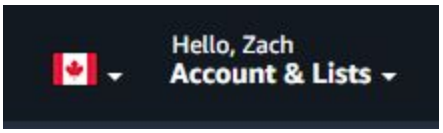
There are two ways to reach the login screen. The first is by clicking the "LOG IN" button on the top right corner of the screen. The second is by creating an order and clicking the "PURCHASE" button. However, these two ways of logging in actually proceed quite differently. If the user logs in the first way, they will subsequently be sent to the payment screen to complete their order. If instead they login the second way, they will remain on the login page, but be shown their current login state. This violates the heuristic "visibility of system status", because there is no way for the user to visually determine which version of the login screen they are on until they finish.



To solve this problem, we recommend adding a hint to the login screen based on the current state. Specifically, if the user reached the login screen as part of the ordering process, give the user instructions such as "Log in to complete your order". Similarly, when the user is logged in, the login page should give the user instructions such as "Click the 'Restaurants' tab to make an order".

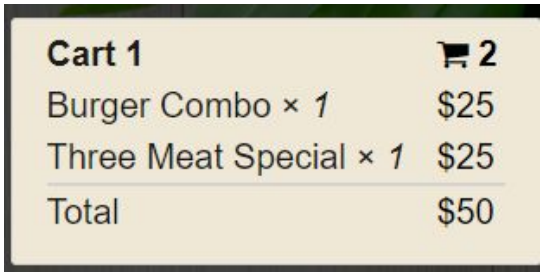
Login Status Not Shown

The only place the user can go to find out whether they are logged in or out is the login page. This violates the heuristic “visibility of system status” because login status may be important information for the user in some circumstances. For example, the user might be confused as to why they are unable to use their previously-saved delivery address if they forget that they are logged in as a guest.



To solve this problem, we recommend adding a clear visual login indicator in a consistent location somewhere in the interface. See the attached screenshot of Amazon’s interface for an example of what we mean.

No Changing Order On Purchase Screen



After selecting the order, the user clicks on the “PURCHASE” button and goes to the payment screen. On the payment screen, suppose they think they do not want an item and want to modify it. At that moment, assuming the interface had a back button (see above), they would go back to the menu screen and modify it. That breaks the user's control and freedom over the interface. Users have to backtrack to the menu screen to modify their order and again go back to the purchase screen to proceed. By backtracking, then going forward again makes the user's task more complicated in terms of achieving their target.

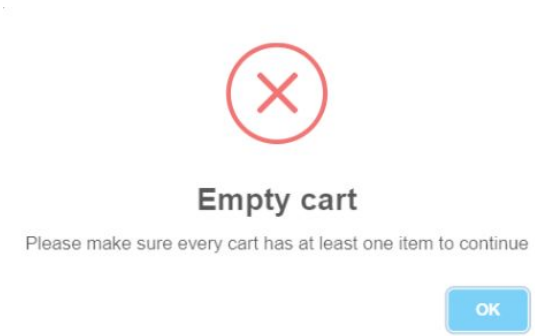
The issue can be fixed, by providing the user a way to remove or change the quantities of the item as well as to modify it, right on the purchase screen. This would give the user more control and freedom through the interface while trying to achieve their goal by doing less work or effort.

Navigation Status

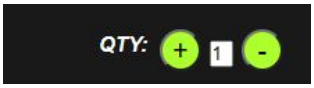
While completing the task the user wants to know where they are and how far are they from their goals. Our interface violates the principle of "Visibility" in the sense that it doesn't have a way for the user to know how many steps they are far from their goal. Without any kind of navigation status, there is a high probability for the user to lose their way.

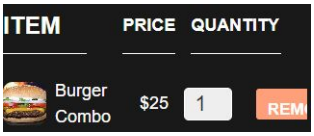
To fix this issue, we recommend adding a progress bar or breadcrumbs to show the progress of the user and highlighting the current stage of the user. So users can know where they came from and where they are going and how far they are from their goal.

Specifying Which Carts Are Empty



Imagine the following scenario: while using the feature of multiple carts, the user selects more carts than they required and only uses some of them while selecting the food items. They then try to checkout with the empty carts. For any interface to maintain its integrity, it should avoid checkout and give an error message with a suggestion to help the user to overcome the error. Our interface violates the “help users recognize, diagnose, and recover from error” heuristic because although it recognizes the error and gives the error message, it does not give a way to recover from it. It does not specify exactly which cart is empty. So the system was unable to offer the solution to the user so that they could solve the problem immediately. The interface also violates the “Feedback” principle due to inadequate feedback.
To fix the problem, we recommend making a change to the error message so in error messages we can also provide a way of recovering from an error. Basically, next time, if the user tries to continue with the empty cart, it pops up an error with a solution specifying "the cart#: number is empty please delete cart# number or add at least one item to that cart".





Quantity Buttons

On the Menu screen, there are two different controls for modifying the quantities of an item: the green "+" and "-" buttons under the menu section, and the standard HTML input controls under the cart pane section. The two different controls violate the "consistency and standards" across the page. These two different controls don't make it obvious to the user that it is the same control to modify the quantities of the item. In addition, the green "+" and "-" button in the menu section violates the "Match between system and the real world" heuristic because in the real world, “+” is usually on the right side and “-” is usually on the left side but, we violate this principle by following the wrong order of the buttons.
To maintain consistency and standards, we should always use the same control for modifying quantities instead of using different controls for the same purpose under the menu and cart section. To solve our violation of "Match between system and the real world", we recommend reversing the order of the quantity buttons so it can match the order that is typical of the real world.

Customizations

When the user is customizing their order, they can input a number per item and customize that item by selecting spiciness and adding special instructions in the provided space. This works well if all of the selected

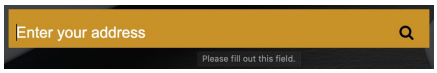
items are to get different customizations but if the user is ordering a number of the same item with the same customizations, they have to go through and edit all of the items. For example, if you selected 10 burgers, each of those 10 burgers would have the same customization unless you changed each one. This breaks flexibility of the system and efficiency of usage. Another issue with the customizations is that if the user wants to add the selected items to their cart, they have to scroll down to the bottom of the customization window, where the “Add to Cart” button is located. This becomes tedious to do if there is a large number of items and it also breaks flexibility and efficiency of usage of the website.

A recommended solution to these issues can be to have one customization for the specified number of items for each Add to Cart window and if the user wants the same item with other customizations, they can input the number of items and have the different customizations in a different Add to Cart customization window.

Invalid Addresses

As the user is going through the process of ordering food from our website, they reach the point where they have to put in the address where their order is to be delivered. Once that is done, they can proceed to selecting a restaurant and continuing with their order. In our design, we had the delivery region to only be a specific area of Winnipeg shown on the map in the “Delivery policy” page. However, if a user types in an address that isn't within that region (or invalid in general), the system does not notify them about whether the address is acceptable or not which violates the feedback and error recovery principles of our interface. In regards to address formatting, there is no guideline or address mask or placeholder used in the address field which would be beneficial in preventing any kind of input format errors. This violates the lack of error prevention of our interface.

To resolve delivery region violations, we recommend an error message coming up and notifying the user that the address they entered is not within the delivery region with an image of the supported area and to resolve the address input errors, we recommend having a placeholder with an input mask inside the address box to help guide the user to use the correct input format.



Help Documentation, FAQ

The website does not have any prewritten guide or tutorial that helps the users to navigate the website or complete any of the tasks they would wish to do. It is a fairly simple interface but assumptions were made that it is simple enough for anyone to manage completing an order

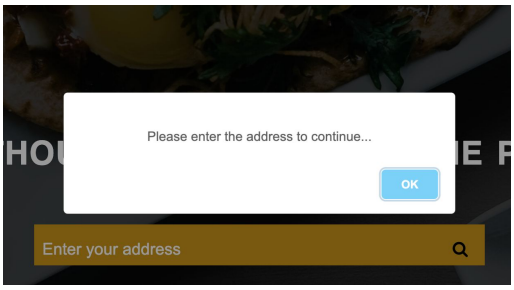
without having to provide a guide for the process. There is a “Need Help” button at the bottom of the page but that would only be useful for when the user has some questions. This is a lack of help and documentation on the website.
To fix this issue, we recommend having some tutorial videos which are easy to follow or a detailed step by step walkthrough of how to go through each of the major processes that a typical user might have to maneuver along with a Frequently Asked Questions page with some answers to common questions.

Showing Errors Earlier

During data entry, like entering the payment details for example, the user types in their information into the payment form and the system checks the format of the data then processes it and continues with the order. However, if there is an issue with the format of the information given, the user is only notified of the issue after they have clicked to submit the form. This makes the process repetitive and tedious as they have to re-enter the information again and fails the lack of error prevention heuristic.
To mitigate this error, we recommend that the system checks the data as it is being entered and gives a notification or an error message when a deviation from the acceptable format occurs.

Can’t Change Order of Restaurants or Food Items

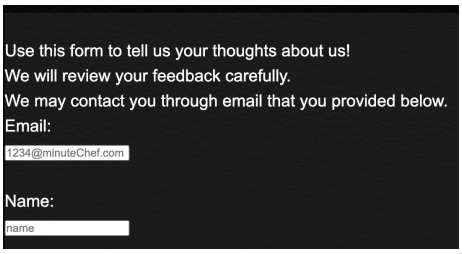
Generally, when dealing with lists that have items which have different attributes for each item, there is a way to change the order that the items in the list are displayed. For the restaurants page on the website, there is a list of restaurants that the user can order from but there is no way to change the order that they are displayed. The restaurants have attributes such as “ratings” and “distance from given address” but are listed in alphabetical order by default. This restricts the users ability to change the order of the listed restaurants if they wish to view them in a different order and this violates the control and freedom heuristic.
To help with this problem, we suggest that the interface incorporates a sorting menu with options that change the order of the listed restaurants to whatever attribute the user chooses.



Error Doesn’t “Feel Like” Error

When the users do not enter an address and proceed to the restaurant page, the system will show the error message. However, the error message box does not look like an error compared to the other error messages because other error messages have the “X with circle“ logo (see picture) but there is no X logo on the address error message. This can be considered a lack of

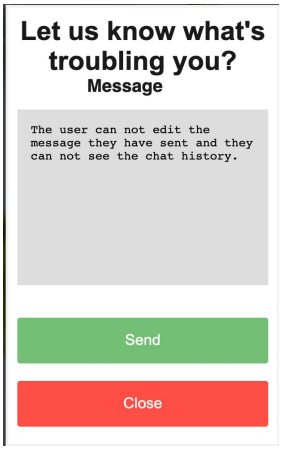
consistency. To solve this problem, we can keep the same design of the error message box everywhere on our website.



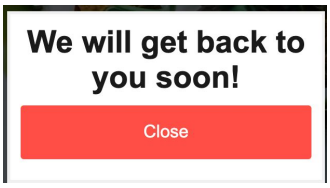
Redundant Email Address Entry

After the users log in to the website and click the feedback page, they will have to input their email address redundantly. This affects the simplicity of the website. We can make the system use the information that is stored already in the account. For example, the system will put the email and name automatically from their account, but we give some option for the users to choose different email or different kinds of contact methods to be contacted.

Can’t Edit/Cancel Help Message



After the users submit a message through the “Need help” section, there is no way to edit or cancel the message. This violates the “Control and freedom” heuristic. Sometimes the users can write unwanted messages and accidently send them. It is important to give more freedom and more controlled experience to the users. In addition, the users can not look back at the history of the chat so if the user asked multiple questions, it is really hard for them to remember all the questions that they have sent.



This is a lack of “recognition rather than recall” for heuristic evaluation.
To solve this problem, we can add an edit/delete button and change the layout like a usual chat application so that the users can check whenever what they have sent to the support team.

Summary

Based on the principles of heuristic evaluation, we found a lot of violations in our system. The violations generated from the process were quite helpful and fixing them would increase the usability of the interface. The things that mainly do not work with the system were firstly not having a back button to go to the previous state. Secondly, the system recognizes errors but does not give a suggestion to recover from an error. Third, taking some sense of control and freedom away from the user where the user has to enter all payment information for multiple carts under one page. There were more violations but they were quite small and easy to solve. On the flip side, we believe our main ideas for the interface went quite well based on the investigation phase; namely, showing everything under the one menu page so the user can know what they are ordering with special instruction in another pane, and supporting multiple carts. Though there were major violations under the interface concerning backtracking, safety, the suggestion of errors and consistency which were quite easy to solve by providing a back button, separating transactions over multiple pages etc. If we were able to solve the issues we found based on the heuristic principles and flow and design of the system, we believe the system is worthwhile in recommending.

Appendix

Zach's Notes

- Two different controls for quantity on the menu: the green "+" and "-" on the left, and the standard HTML number input on the cart pane
    - Breaks "consistency and standards"
    - Should make them consistent, one way or the other
- After going to the purchase screen, there's no way to go back to change your order
    - Breaks "user control and freedom"
    - Should add a back button
- Related to the last one: on the purchase screen, there's no way to remove or change quantities of items
    - Breaks "user control and freedom"
    - Should support this for convenience
- All users have to enter their payment information on the same screen
    - Breaks "flexibility and efficiency of use"
    - Should email the bills to each user separately
- Two different login states with no visual distinction
    - When you click "LOG IN" on the top right corner of the screen, the system takes you to the login screen. When you login, you stay on the login screen, which now tells you you're logged in. however, when you go to purchase an order, the system *also* takes you to the login screen, but when you finish, it sends you to the payment page. This could be confusing to some users.
    - Breaks "visibility of system status"
    - Should display a message in the latter message, like "Log in to complete your order"
- If you want to order a bunch of the same item with the same customizations (spiciness and notes), you have to manually enter the customizations each time
    - Breaks "flexibility and efficiency of usage"
    - Not sure how to fix this. Maybe in the customization pane, just have one customization, instead of one for each quantity? Then if you selected, for example, 10 burgers, each of those 10 burgers would have the same customization.
- If you select a large quantity of an item, you have to scroll way down on the customization pane to add it to your order
    - Breaks "flexibility and efficiency of usage", and arguably "recognition rather than recall"
    - Solution would be the same as the above. Just have one customization per "add to cart" action
- If you are scrolled down in the page when you navigate to the next page, it will start off scrolled too far down

- Breaks “Flexibility and efficiency of use”
- When the user navigates to a new page, the system should automatically scroll to the top

Zeel’s Notes

Violates control and freedom

- There is no back button on the website so users can not go to the previous state once they move to the next state.

Breaks error prevention

- When clicking on the purchase button and the pay button do not ask for the confirmation page in a sense that does not warn the user when completing the task which can cause the big errors.

Aesthetic and minimal design

Combos on the menu page should be placed in the category by making the new field of category.

Help user recognize, diagnose and recover from error

- When a user does not enter an address error message coming in blue color doesn't feel like an error.

Tinotenda's Notes

After submitting a message on the "Need help" section, there is no way to edit or cancel the message

- Control and freedom
- A message chat history with ability to edit messages.

If the address entered is not in the area delivered to, there is no feedback

- Error recovery, feedback
- Error message stating address is not in supported region

There is no way to change the order of the restaurants (according to distance or ratings)

- Control and freedom
- Restaurant sort option

After trying to purchase with an empty cart, the system simply states that there is an empty cart but does not say which one is empty

Feedback, error recovery, simplicity

More detailed error message stating which cart is empty

When submitting feedback after logging in, having to input email address again

- Simplicity,
- If you are logged in, system should recognize email address and name of user logged in and automatically use those details

You cannot change restaurants without going back to the screen where you input your address

- Control and freedom
- Add a back button

+ and - buttons for increasing number of items to add to cart, + is usually on the right side

- Consistency compared to everywhere else even though it was like that throughout the interface
- + on the right and _ on the left

If you go back to the home page, you have to repeat your whole order again, even if there were items in the cart

- Minimize memory load, control and freedom
- Resume order and keep address in address bar

Sangmin’s Notes

The quantity button

- The quantity button at the combo’s sections starts with the “+” button and ends with “-” . However, the quantity button at the cart’s section starts with “-” first and ends with “+”
    - This violates consistency.

Log in status

- Anywhere in the site, there is nothing informing if the user is logged in or logged out until they reach the login page. The users will be confused by this.
    - Lacks of visibility of system status

Log in (preventing errors)

- The users will identify the error after they insert the data. It is always better to know the error before it happens.
    - Lack of error prevention.

Help documentation

- There is no documentation for helping/guide users about using the website. There is a “Need help” button on the right bottom but that is the only one for users to get information when they have some questions. It is better to have a page that shows Frequently Asked Questions.
    - Lack of Help and documentation

Address box

- There is no example of address form on the placeholder for the input box, and no error messages for awkward inputs.
    - Lack of error prevention.

Payment page

- The users can not modify their orders when they reach the payment page.
    - Lack of user control and freedom.