

# **FarmerEx Platform**

## **UX Report**

By: Olivia Kiklica, Clark Mahaffey, Ashley Kelly, & Bobby Bloomquist

JDF-1132

*For*

Karlyn Wilson,

The Atlanta Mobile Market

April 11, 2021

# Table of Contents

List of Tables .....	2
List of Figures .....	2
Introduction.....	4
Prototype.....	5
<i>Producers</i> .....	5
<i>Consumers</i> .....	10
Methods.....	11
Findings and Recommendations .....	13
Revision of Prototype.....	14
<i>Scenario 1</i> .....	14
<i>Scenario 2</i> .....	15
<i>Scenario 3</i> .....	15
<i>Scenario 4</i> .....	17
<i>Scenario 5</i> .....	17
Appendices .....	18
<i>Appendix I: Heuristics Evaluation by Team #1138</i> .....	18
Part 1: Task Evaluation.....	18
Part 2: General Heuristic Evaluation of Prototype Interface.....	22
<i>Appendix II: Heuristics Self-Evaluation</i> .....	26
Part 1: Task Evaluation.....	26
Part 2: General Heuristic Evaluation of Prototype Interface.....	32
<i>Appendix III: Heuristics Evaluation by Course Professors</i> .....	35

## List of Tables

Table 1: Scenarios and Tasks .....	11
------------------------------------	----

## List of Figures

Figure 1: Producer Dashboard .....	5
Figure 2: Producer Pending Sales Screen .....	6
Figure 3: Producer Payment Screen .....	7
Figure 4: Producer Storefront Screen .....	8
Figure 5: Producer Decline Order Screen.....	8

Figure 6: Producer View Order Screen.....	9
Figure 7: Producer Storefront Screen .....	10
Figure 8: Revised Payment Authorization Screen .....	14
Figure 9: Revised Order Confirmation Screen .....	15
Figure 10: Revised Order Decline Screen .....	16
Figure 11: Revised Consumer Screens.....	17

## Introduction

Our team is working with the Atlanta Mobile Market to develop FarmerEx, a platform intended to connect local farmers with communities in need of fresh produce. Food insecurity – that is, a lack of access to affordable and nutritious foods – has historically been an issue in low-income communities; the COVID-19 pandemic has only compounded the problem, as our user research revealed that consumers are increasingly wary of shopping in enclosed spaces. The consequences of food insecurity are far-reaching, and can potentially compound into long term diet-related conditions such as obesity and diabetes – conditions exacerbated by a lack of access to fresh fruit and vegetables.

FarmerEx aims to mitigate food insecurity, food waste, and chronic health issues resulting from prolonged malnutrition by increasing access to local fresh produce through two easy-to-use interfaces. While the producer section of our intended userbase will access the platform via a desktop browser interface, the consumer audience will instead use a mobile interface.

The choice to develop individual interfaces for each audience stems from two sources: projected feature complexity, and technological access and literacy as defined by our user research. Producers are more likely to be processing large amounts of data (e.g. managing available produce types and volume, sorting through customer requests, uploading/downloading spreadsheets of availability and profit, etc.); a desktop interface is therefore appropriate for this more intense computing.

Consumers, however, use a mobile interface because the process of simply *buying* produce is not resource-intensive: all it requires is an account and a valid form of digital cash. Additionally, it is an unfortunate reality that food insecurity is only one facet of poverty: members of lower income communities are generally less likely to pursue postsecondary education. As a result, a presumed lower level of digital literacy requires us to design a clean and simple mobile interface – our consumers are not expected to be power users. Additionally, a portion of our user research (gathered from the ACS 2018 Community Survey) indicated that if a household in our targeted area owned only one piece of technology, it was more likely to be a smartphone than a desktop computer.

With these insights in mind, we designed each interface to be easy to use, yet tailored to each audience's needs. We then evaluated our designs internally, using Jakob Nielsen's Usability Heuristics, as well as externally with another group; we showed them our list of intended scenarios and tasks and had them navigate our prototype with the intent of completing these tasks. Finally, we demonstrated our prototype to our instructors and client. We received valuable feedback from all these sources (particularly regarding interface navigation) which we then integrated into our revised prototype.

After our research-backed revisions, our team believes that we have created intuitive user interfaces for both the consumer and producer audiences on the FarmerEx platform.

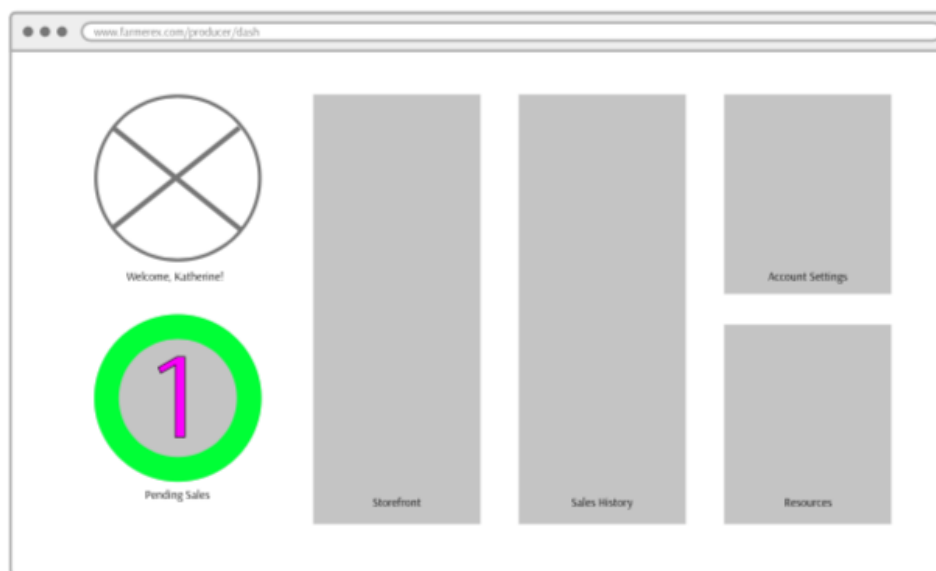
## Prototype

This section details the core screens of our initial prototype, on both the producer and consumer end.

### Producers

The producer web interface is an orderly grid of options: Storefront (what goods the consumer sees), Sales History, Account Settings, and Resources. On the left hand side the producer's profile picture is displayed. Because sales must be approved by producers before the transaction can be completed, there is also a 'Pending Sales' icon on the left, underneath the profile picture.

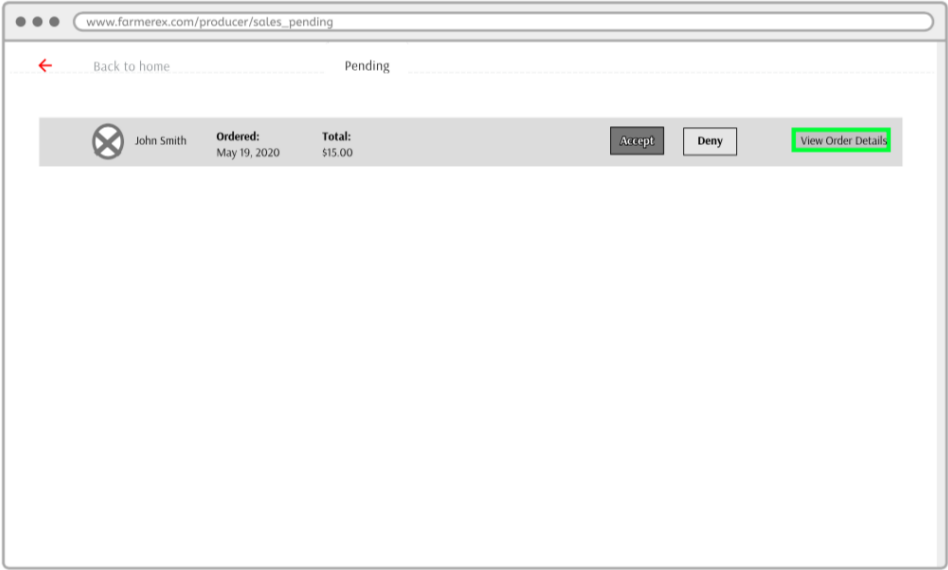
*Figure 1: Producer Dashboard*



We chose to make the 'Pending Sales' button distinct from the others because of its importance to economic movement within the marketplace; to put it simply, retaining consumers on our platform requires expedience on the producers' part – consumers are more likely to stop using the app if each sale takes a significant amount of time to complete.

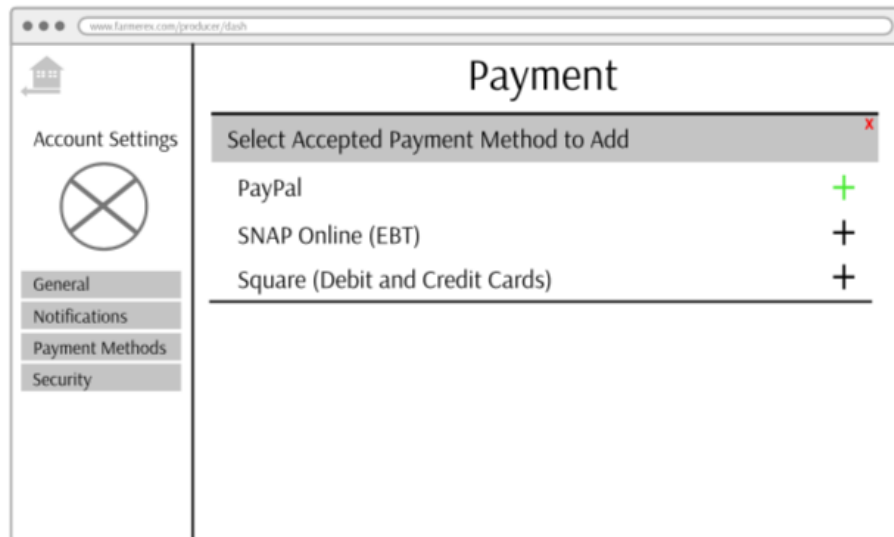
Clicking on the 'Pending Sales' button leads the user to the 'Pending Sales' screen, which is likewise organized in a grid style (albeit single column). This clean interface allows the user to accept or decline orders without further investigation, or to view order details in order to learn more about the consumer's order and location. Choosing 'Accept' or 'Deny' will lead to the confirmation or cancellation screen, respectively; however, selecting 'Deny' will prompt the user to enter a reason from a list as to why the order is being denied.

Figure 2: Producer Pending Sales Screen



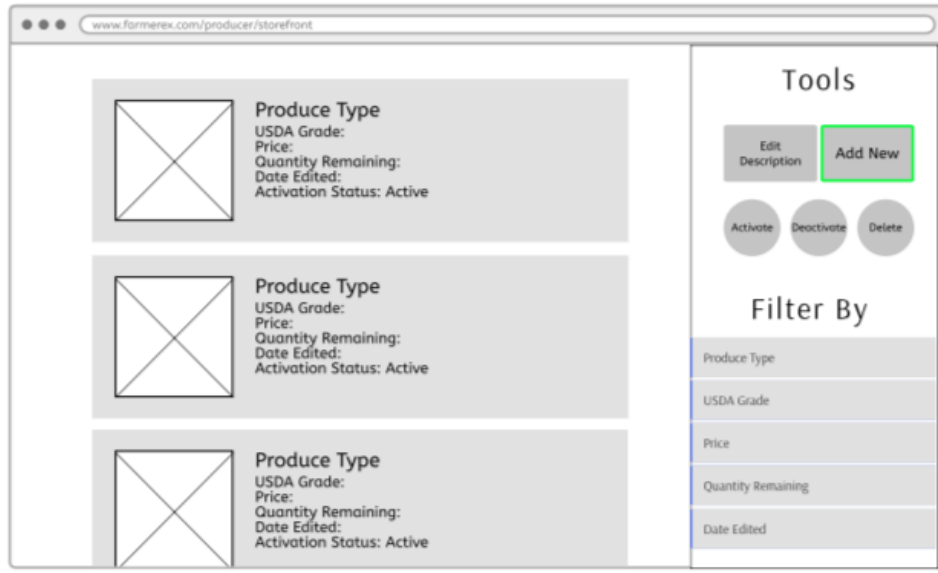
The 'Account Settings' screen offers the user the ability to adjust their notifications, payment, and security settings, as well as more general settings (such as the user's profile picture and display name). Here we designed the 'Payment Methods' screen, which allows the user to connect their point-of-sale accounts from PayPal, SNAP Online, or Square, in order to process consumer payments.

*Figure 3: Producer Payment Screen (A Subset of Account Settings)*



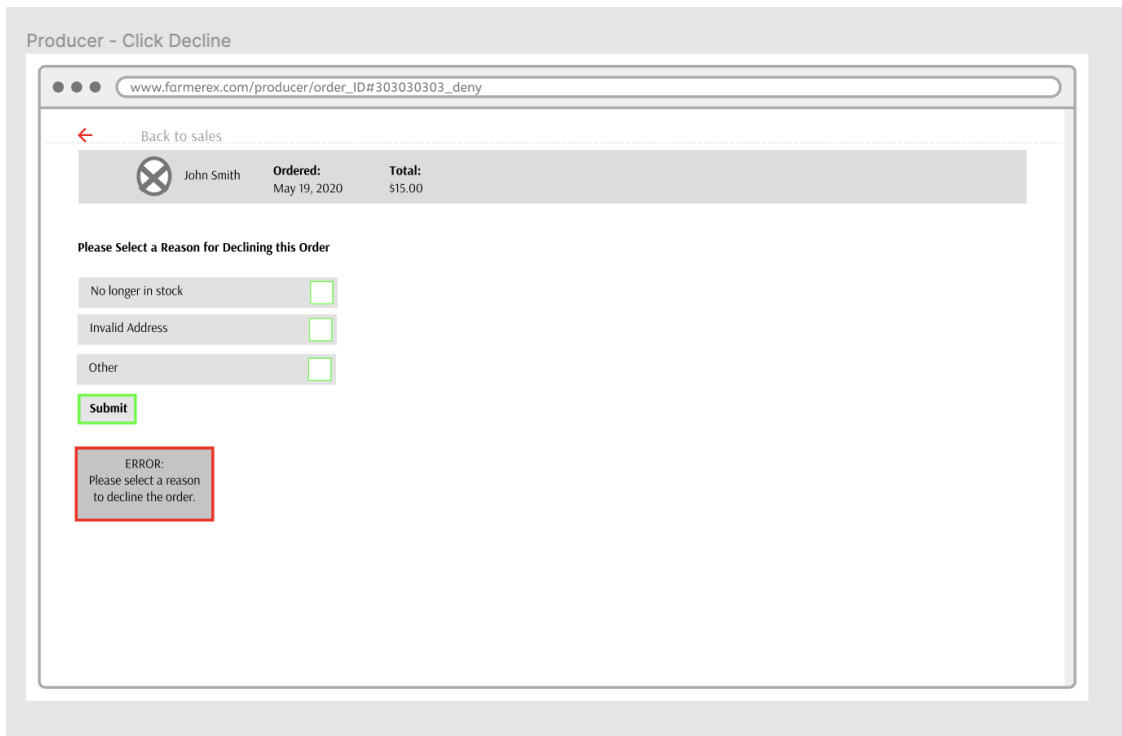
The 'Storefront' button leads to the 'Storefront' page, where users can edit their description of listed produce, add a new produce listing, activate or deactivate a listing (which essentially adjusts visibility to the consumer), and delete any listing. When adding a listing, the user can choose the produce type from a dropdown; custom types are not allowed in order to reduce variability in listings and make searching for goods easier on the consumer end. Editable fields include USDA Grade, USDA organic status, price, quantity remaining, and activation status; users can filter by any of these fields.

Figure 4: Producer Storefront Screen



When a producer declines an order, the user can give the consumer feedback about why their order was ultimately denied. The user can select that the produce was no longer in stock, the address was invalid, or something else prohibited the sale. The user must click the submit button for the cancellation to be processed, otherwise they will get an error message.

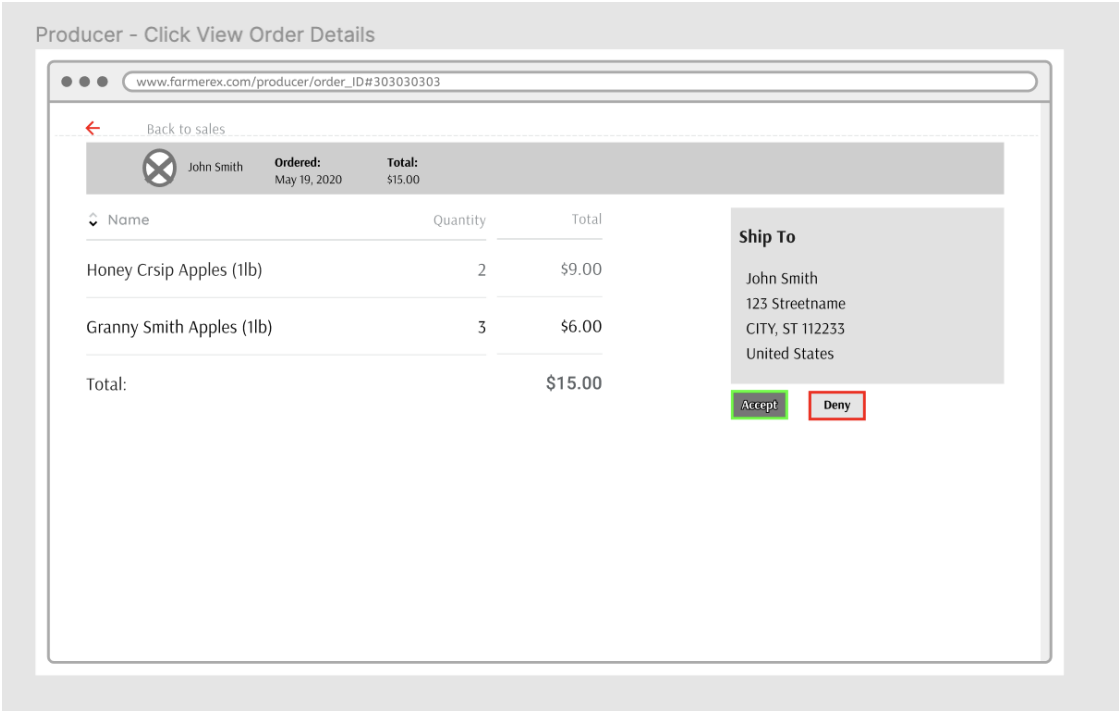
Figure 5: Producer Decline Order Screen





When a producer selects to view order details on the 'Producer Pending Sales Screen', they are redirected to the 'Producer View Order Screen' shown below. This screen offers more insight into the order they have selected such as the order's individual items as well as the pricing and quantity of those items. The location of the consumer that made the order is also shown. Here, the producer can consider this additional information to evaluate if they wish to accept or decline the order. The producer is given the same options to accept or deny the order that were present on the 'Producer Pending Sales Screen'. Similar to the option on the previous screen, these will also redirect to a confirmation screen or a screen requesting a reason for the decline respectively.

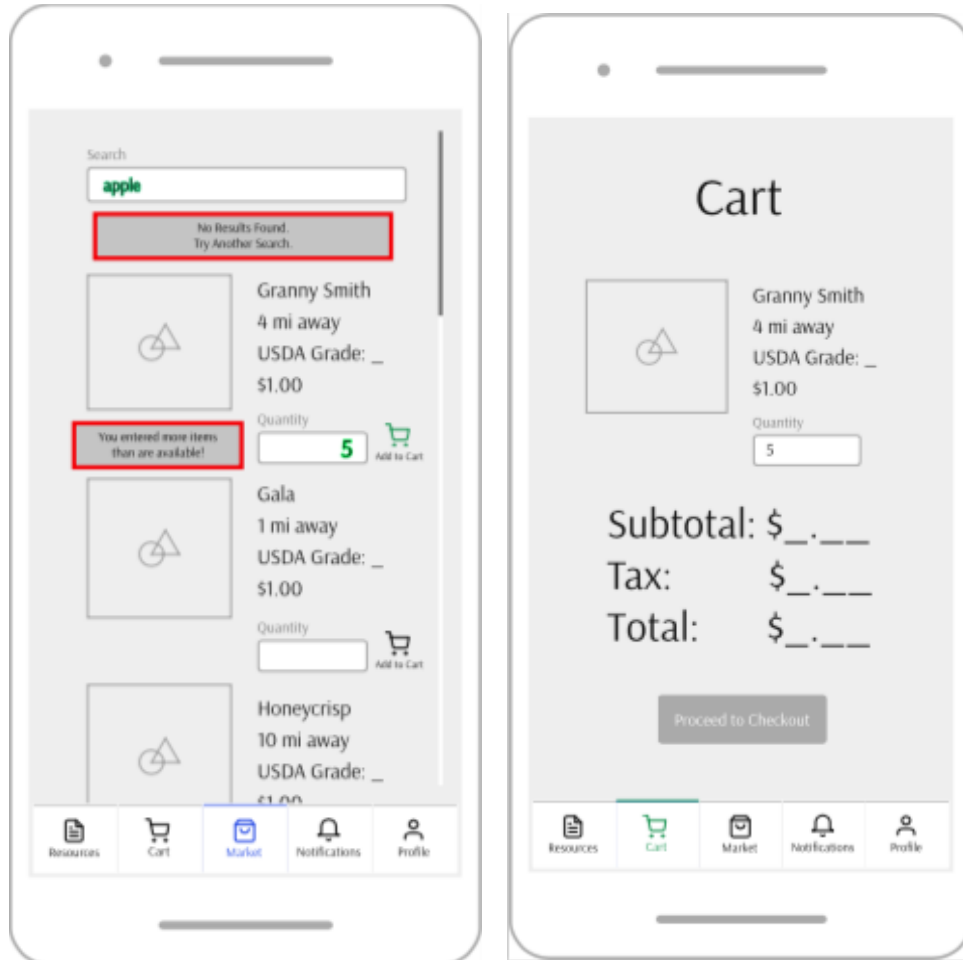
Figure 6: Producer View Order Screen



## Consumers

The consumer application is less complex than the producer application. For the sake of brevity, two main screens are shown here. In the first, the user searches for a produce type and is presented with what is available in their area. Clicking 'Add to Cart' will add the selected produce in the selected quantity to their cart; the cart can then be viewed at any time, along with subtotal, tax, and total expenditure, and allow the user to proceed to the checkout screen.

Figure 7: Producer Storefront Screen



## Methods

The evaluations of our prototype were conducted using Jakob Nielsen's 10 Usability Heuristics for User Interface Design, across three separate groups: two external evaluations were performed by JDF-1138 (a fellow project team) and our professors, while a third internal review was conducted by individuals within our team. The same five scenarios (three producer scenarios on the web application, and two mobile-based consumer scenarios) were presented to each group.

*Table 1: Scenarios and Tasks*

Scenario #	Scenario Description	Tasks
Producer Scenarios		
1	A producer has just registered for the site, the home dashboard is open, and they want to authorize PayPal as an accepted payment method.	<ul style="list-style-type: none"> <li>i. Open account settings.</li> <li>ii. View payment settings.</li> <li>iii. Add new account to receive payments with.</li> <li>iv. Select PayPal from list of payment methods.</li> <li>v. Log in to PayPal and authorize PayPal account.</li> <li>vi. Exit and view all authorized accounts.</li> </ul>
2	A registered producer has logged into the site, the home dashboard is open, and they want to list a new item for sale.	<ul style="list-style-type: none"> <li>i. Open Storefront.</li> <li>ii. Open 'Modify Storefront.'</li> <li>iii. View produce folders.</li> <li>iv. Select 'Add Item.' Select item type to add.</li> <li>v. Tag item attributes.</li> <li>vi. Fill out quantity.</li> <li>vii. Save storefront changes.</li> </ul>
3	The registered producer has logged into the site, the home dashboard is open, and they want to view and evaluate any pending sales.	<ul style="list-style-type: none"> <li>i. Open 'Pending Sales.'</li> <li>ii. Open a pending sale to review information.</li> <li>iii. Confirm or deny sale.</li> <li>iv. If deny, select reason.</li> </ul>
Consumer Scenarios		
4	A consumer is looking to find and add produce to their cart, so their total is under 10 dollars. They have the Mobile Market Place open.	<ul style="list-style-type: none"> <li>i. Search for items.</li> <li>ii. Add items to cart.</li> <li>iii. View cart.</li> </ul>
5	A registered consumer has Mobile Market Place	<ul style="list-style-type: none"> <li>i. View cart.</li> </ul>

	open, and they want to check out and buy the items currently in their cart.	ii. Choose payment method. iii. Check out.
--	---	---

These scenarios tested what we considered to be the cornerstones of the product. From the producer's side, fundamental functions include the ability to accept payments, list items, and accept sales. Likewise, from the consumer-side fundamental functions include the ability to search for items, add desired items in the correct quantity to the shopping cart, and checkout their cart. In order to test the efficacy of our prototype in completing these basic tasks, we undertook a walkthrough of the prototype with each of the three previously mentioned parties.

Due to current COVID-19 distancing guidelines, walkthrough evaluations took place on Microsoft Teams using the screenshare function. In each walkthrough we gave the other party each of the tasks defined in Table 1 and asked them to navigate the relevant interface in order to complete the task. On each screen, our evaluators were encouraged to ask questions and point out any perceived flaws or instabilities in the design. In the case of the evaluation by the other team, we received a completed form from them with detailed evaluations of our designs (as defined in Jakob Nielsen's aforementioned heuristics) – our team completed a similar form for our internal evaluation. During the instructor evaluation, we took notes as they commented on the interfaces.

The findings of these evaluations are reported in the next section.

## Findings and Recommendations

According to the three performed evaluations, our initial prototype had several strengths. Team 1138 reported that their user experience was logical and smooth, and the designs were clear and consistent. They also praised our error prevention, feedback, and our design's prioritization of recognition over recall; in particular, they appreciated how the design would not confuse the user into conflating different functions (e.g, clicking on 'Account' when they want 'Storefront'). Our self-evaluation revealed our distinct 'Pending Sales' icon to be a strong component of the producer interface. The professor evaluation praised our usage of color.

However, these evaluations also revealed several flaws in our design. The severity of these errors is sorted into three categories, listed here in descending order:

1. **Critical** – The error(s) prevent the user from completing the task.
2. **Serious** – The error(s) cause the user to face increased difficulty in completing the task, but completion is *not* impossible.
3. **Minor**: The error(s) pose little to no hindrance to task completion.

The evaluations revealed our design as suffering several overall errors, though luckily only one critical error was identified.

### *Critical Errors*

One critical error was found in the consumer cart screen. Once a user added an item to their cart, they had no way of removing it. This presented the critical issue of preventing the user from having control over customizing their cart and clearly violates the 'Error Prevention' heuristic.

### *Serious Errors*

Some errors were serious. The instructors pointed out that we needed a way to deal with multiple people buying the same item at the same time. Users will become quickly frustrated if they think they're buying available produce, when in reality someone else has 'taken' their goods, causing their order to decline.

We also needed a way to get better feedback from producers on why they denied a sale. The "other" option as initially conceived did not have a text box for custom responses. Not giving the user the ability to provide open-ended feedback undermines the purpose of the checkbox survey, which is to collect data on *why* a user is declining to see if a stumbling block can be removed on the platform's end. In order to improve this screen's ability to collect information, our evaluators recommended we add a textbox. All this being said, this error, as it stands, does not inhibit the user from completing a denial of an order: it only inhibits useful feedback. Therefore, this is a serious but not critical error.

The final issue on the sale denial screen is the lack of cancel button, which does not give the user enough control. The user can currently cancel by hitting the back arrow; however, it might be more intuitive to include a cancel button as well. This error does slightly impact the user's ability to evaluate sales, so it is a serious, but not critical flaw. Team 1138 pointed out that there should be a warning before the producer finally denies a sale.

### *Minor Errors*

Our prototype had sundry minor errors. The payment information screen had a capitalization error on PayPal, and Honeycrisp apple was misspelled. The instructors asked us to remove the exclamation points from our error messages, so the user isn't annoyed. Within our own team, we realized that we should add better headings and titles to our screens.

Our initial design did not display feedback messages to let users know an action had happened. For instance, the platform lacked a confirmation message when a producer adds an item for sale or accepts an order. For legibility purposes, the producer item sale screen needed concrete examples of produce instead of placeholders like "Produce A" and "Produce B". For searching produce, the instructors recommended we decide if the search would happen via keywords or tags. Numeric text fields would need to be replaced with drop down menus.

On the producer decline sale screen, there was a lack of clarity in the "Invalid address" option. The purpose of this option was intended to specify that location is the reason for declining the order. However, this reason is better handled internally with geofencing and address validation and/or rephrasing the option to explain that the location is "Outside delivery area". This, again, does not inhibit the user from completing the task, so it is only a minor error.

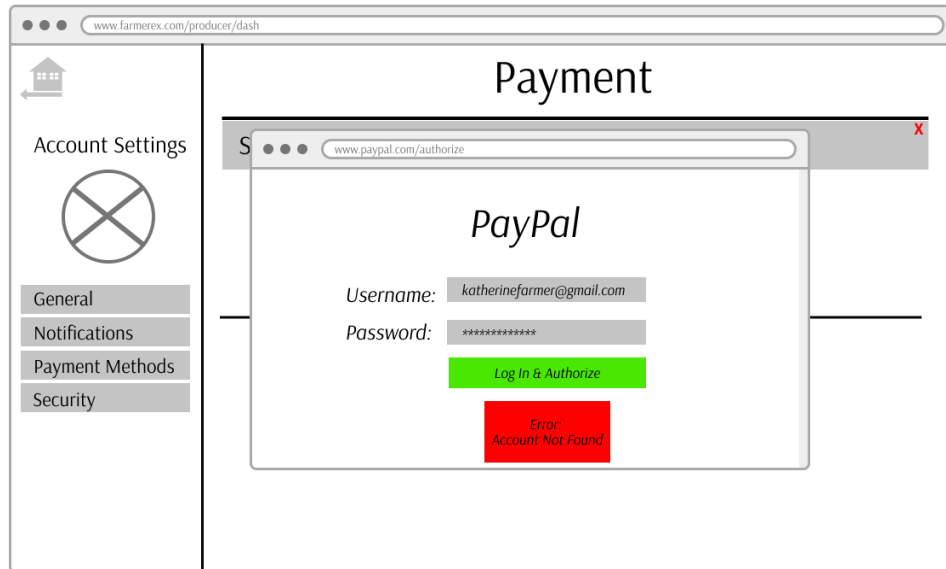
### *Revision of Prototype*

Once we received and aggregated all of our feedback, we set about correcting the errors, scenario by scenario.

#### *Scenario 1*

The producer flow for adding accepted payment methods required few modifications between the initial draft and the revised prototype. Only two small modifications were made, both minor errors corrected in the interest of branding and user-friendliness. The first change concerns the brand name 'PayPal,' which was incorrectly displayed as 'Paypal' in two instances; those instances were corrected. The second change concerns the display of errors. Whereas in the initial draft error messages were more aggressive, reading 'Error!', this message was changed to 'Error:' (followed by the error) in order to be less intimidating to users without a computing background.

*Figure 8: Revised Payment Authorization Screen*



### *Scenario 2*

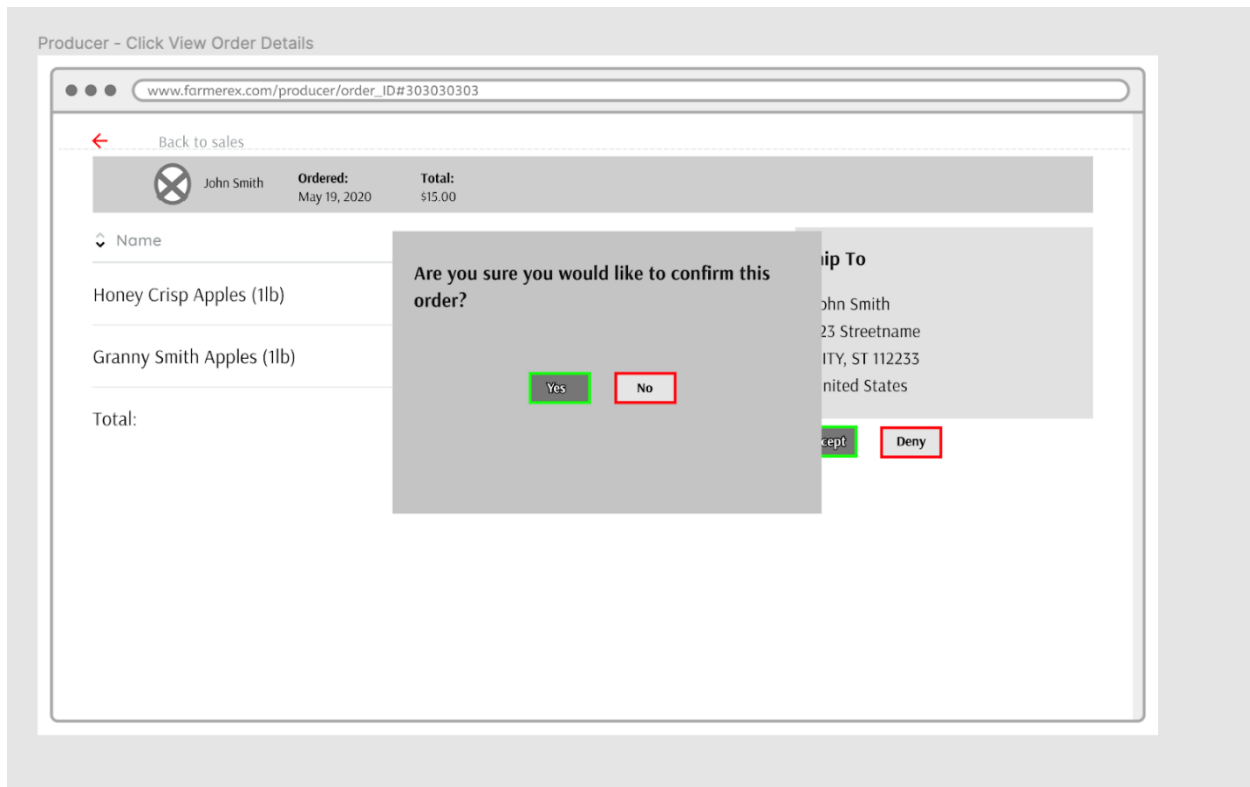
As with the first scenario, few modifications needed to be made to the original draft of Scenario 2 (which entails a producer listing a new item for sale). Two of our changes were minor, and both were to give the user a better understanding of what modifications occurred while editing the Storefront.

The first addition was a feedback message to let the producer know that the item they chose was successfully added to their storefront. This message helps with the visibility of system status in letting the user know where they are in the application. The next interface change replaced all placeholder names (e.g. Produce A) with real produce names, like apple. The original draft was confusing to reviewers, who found that using variable names in place of produce names (e.g. 'Produce A' rather than 'Apple') negatively affected the interface's legibility.

Finally, a serious change was made in order to prevent duplicate items from being added to the storefront. Adding a duplicate item can be very confusing and easy to do for any user. To fix this issue, an error message was added to let the user know that they cannot create a duplicate item.

### *Scenario 3*

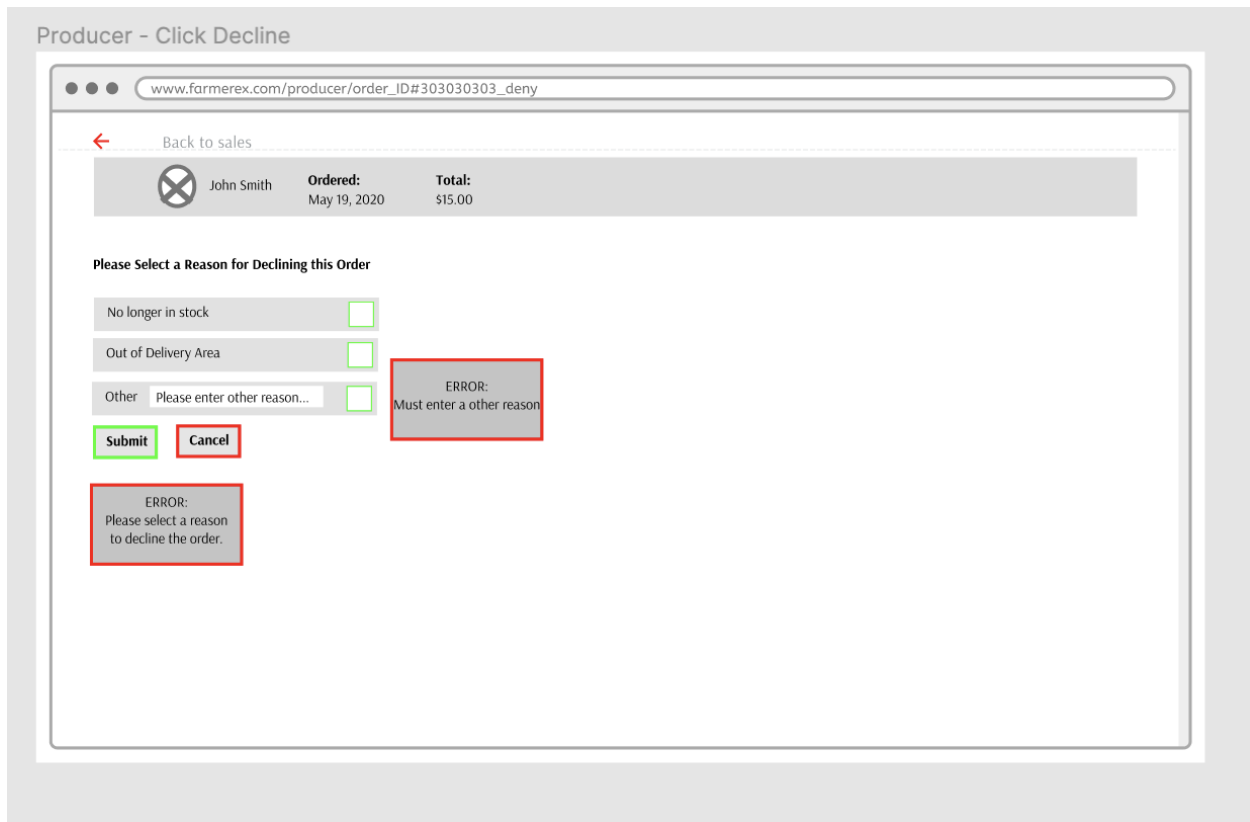
*Figure 9: Revised Order Confirmation Screen*



These changes to this screen fix the typo of "Honey Crisp Apples" and the lack of confirmation screen when confirming an order. Fixing the typo elevates the professionalism of the prototype and makes the application seem more trustworthy. The second flaw that was fixed is seen with the addition of another level of confirmation when confirming an order. By having this additional option menu when clicking 'Accept,' the user can no longer accept an order with a single click, which will work to minimize accidental confirmations.

*Figure 10: Revised Order Decline Screen*





This screen's revisions fix two flaws. The first is the lack of a cancel button for denying an order: by adding a cancel button here, the application gives the user clear control and tools to correct mistakes. Therefore, frustration and panic when the 'Deny' button is accidentally clicked will be minimized. Additionally, a textbox was added to the 'Other' option in order to give the user the ability to provide more open-ended feedback on why they canceled their order. This will allow us to collect better data on the user and give the user a better avenue to express feedback.

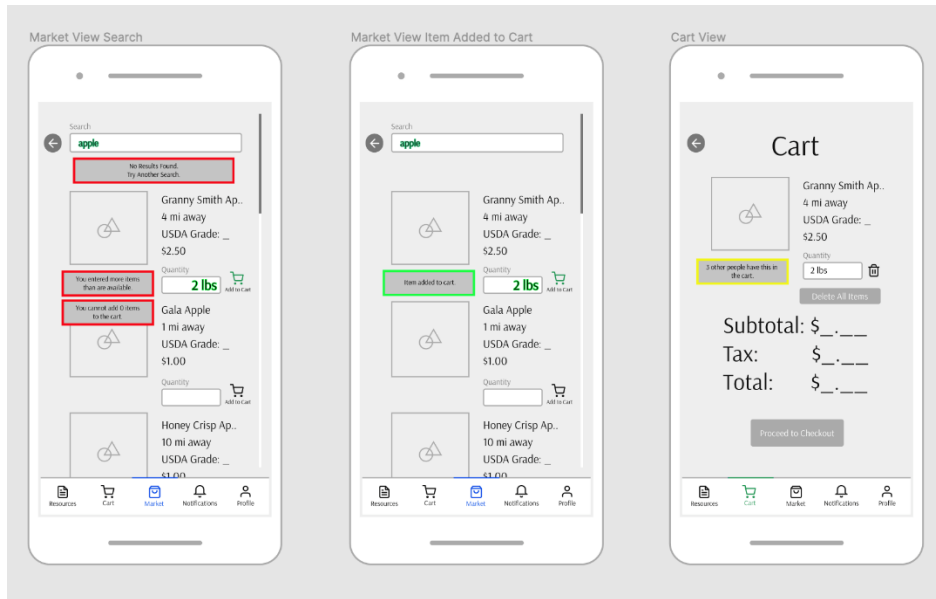
#### *Scenario 4*

While our team decides how we will implement the search function with keywords or tags, the names of the items on screen include the actual keyword "apple". When the user entered a quantity of items, there was already an error message for going over the maximum amount, but an error message was added for entering 0 items. A helpful message was added indicating that an item had indeed been added to the cart. Third, to warn customers, a message was added to the cart screen to indicate that other people were looking to buy the same product simultaneously. The user has more control over their experience with a new back button to take the user to the previous screen. The user can now delete individual items from the cart or delete all the items at once.

#### *Scenario 5*

When checking out, a helpful link was added beneath the SNAP/EBT payment option for users to discover their eligibility.

*Figure 11: Revised Consumer Screens*



## Appendices

### *Appendix I: Heuristics Evaluation by Team #1138*

#### Part 1: Task Evaluation

By Team # 1138: Tam Truong, Reynold Kyaw, Euseong Jeoung, Wiley Gray

Part 1: Task Evaluation

Heuristic	<b>Evaluation</b> As each task is performed, enter in the space below your observations and evaluation of the degree to which the relevant heuristic has been satisfied. Note: Not every heuristic may be relevant for each task. Use as much space as you see fit.				
	<i>Task 1</i>	<i>Task 2</i>	<i>Task 3</i>	<i>Task 4</i>	<i>Task 5</i>
	<i>Producer adds a payment processor.</i>	<i>Producer wants to list a new item for sale.</i>	<i>Producer Accept/Decline orders</i>	<i>Registered consumer loads their virtual cart with items.</i>	<i>Registered consumer has marketplace open and wants to check out.</i>
1. Visibility of system status	<i>Looks good, everything is very visible and clear.</i>	<i>Feedback message that says "Produce C added" would be helpful.</i>	<i>Everything is visible and clear.</i>	<i>Everything is visible and clear.</i>	<i>Everything is visible and clear.</i>
2. Match between system and the real world	<i>Satisfied, logical and smooth flow.</i>	<i>Satisfied, no clear problems.</i>	<i>Satisfied, no clear problems.</i>	<i>Satisfied, intuitive workflow.</i>	<i>Satisfied, no clear problems.</i>
3. User control and freedom	<i>Users can navigate to another page by clicking on the sidebar.</i>	<i>User can easily select the item they want to add to start selling.</i>	<i>User is free to accept and decline orders easily with clear controls.</i>	<i>User is free to choose any product they desire.</i>	<i>User can easily control when to check out and what method to use.</i>

4. Consistency and standards	<i>Satisfied, looks good, no confusion in wording.</i>	<i>Satisfied, does not look too complicated to understand.</i>	<i>Add a text box under "Other" in the "Deny" page. On the side, how would the seller know if an item is out of stock? Is there a tracking system that helps them to do that?</i>	<i>UI is very clean and consistent regarding the products and prices. Add a small notification when the user clicks "add to cart" so they know that it's in their cart.</i>	<i>User still needs a way to delete an item from their cart.</i>
5. Error prevention	<i>Satisfied, the scenario has its own error prevention as cannot logged in. Add a warning box when the user clicks on delete current Payment method.</i>	<i>Account for duplicating products. Ask the user if they want to add the same product again even if it exists.</i>	<i>Add a warning box when user clicks "Deny" or "Confirm".</i>	<i>Add removing items in the cart. Add the warning box when removing.</i>	<i>Satisfied, the wireframes represent the situation well with the error in payment.</i>
6. Recognition rather than recall	<i>Satisfied, cleared labeled and buttons that explains their functions well.</i>	<i>Satisfied, steps are cleared, and the process to get there is well described and easy to follow.</i>	<i>Satisfied, easy to follow the scenario and wordings are matching the description.</i>	<i>Satisfied, easy to navigate on mobile app.</i>	<i>Satisfied. Looks perfect, cleared, and self-explanatory labels.</i>
7. Flexibility and efficiency of use	<i>User can add payment method just by clicking plus button and put in payment method information.</i>	<i>Very seamless, quick navigation. Able to switch products with just a click.</i>	<i>The producer can accept or decline orders within 3-page navigations from the landing page, very good.</i>	<i>Satisfied; Search feature speeds up process.</i>	<i>Satisfied, easy and quick checkout button.</i>
8. Aesthetic and	<i>Looks sleek</i>	<i>Looks sleek</i>	<i>Looks sleek</i>	<i>Looks sleek</i>	<i>Looks sleek</i>

<b>minimalist design</b>	<i>and modern.</i>	<i>and modern.</i>	<i>and modern.</i>	<i>and modern.</i>	<i>and modern.</i>
<b>9. Help users recognize, diagnose, and recover from errors</b>	<i>The possible error which is 'Account Not Found' is stated.</i>	<i>The possible error which is 'All fields must be filled out' is stated.</i>	<i>A possible error when they did not choose any reason for decline is stated.</i>	<i>A possible error when user entered more items than available quantity is stated.</i>	<i>If quantity entered is zero, there should be a dialogue box correcting the user.</i>
<b>10. Help and documentation</b>	<i>N/A</i>	<i>N/A</i>	<i>N/A</i>	<i>N/A</i>	<i>One link to SNAP/EBT help</i>

## Part 2: General Heuristic Evaluation of Prototype Interface

**Part 2: General Heuristic Evaluation of Prototype Interface**

<b>Heuristic</b>	<b>Evaluation</b> In the space below, enter your observation and evaluation of the degree to which the heuristic has been satisfied. Use as much space as you see fit.
<b>1. Visibility of system status</b> <ul style="list-style-type: none"> <li>• Always keep users informed about what is going on.</li> <li>• Provide appropriate feedback within reasonable time.</li> </ul>	<i>The site is very clear for all scenarios, each task has a clear next step to any user all the way through to task completion.</i>
<b>2. Match between system and the real world</b> <ul style="list-style-type: none"> <li>• Speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms.</li> <li>• Follow real-world conventions, making information appear in a natural and logical order.</li> </ul>	<i>The site uses short, understandable labels for each box/button. Both producers and consumers should be able to easily understand their actions.</i>
<b>3. User control and freedom</b> <ul style="list-style-type: none"> <li>• Users often choose system functions by mistake.</li> <li>• Provide a clearly marked "out" to leave an unwanted state without having to go through an extended dialogue.</li> <li>• Support undo and redo.</li> </ul>	<i>The site is very easy to use and grants the user a lot of control over what is taking place. All types of users have full freedom to navigate and interact with whatever they see fit using the tabs on the left and at the bottom for producers and consumers respectively.</i>
<b>4. Consistency and standards</b> <ul style="list-style-type: none"> <li>• Users should not have to wonder whether different words, situations, or actions mean the same thing.</li> <li>• Follow platform conventions.</li> </ul>	<i>This site is consistent and followed the conventions that allows users when using this site to have an easy time to familiarize the platforms because everything is in a coherent order where the user can understand the flow of the site.</i>
<b>5. Error prevention</b> <ul style="list-style-type: none"> <li>• Even better than good error messages is a careful design which prevents a problem from occurring in the first place.</li> </ul>	<i>This site needs to add more error and warning messages to make sure the user knows or want what he/she is doing.</i>

<p><b>6. Recognition rather than recall</b></p> <ul style="list-style-type: none"> <li>• Make objects, actions, and options visible.</li> <li>• User should not have to remember information from one part of the dialogue to another.</li> <li>• Instructions for use of the system should be visible or easily retrievable whenever appropriate.</li> <li>• Buildings are all visible and clickable</li> <li>• Experienced users can directly check on the buildings they are interested in by clicking on them</li> <li>• Recent searches feature is useful for users that use the same buildings regularly</li> </ul>	<p><i>This site is user friendly and cleared with actions and visible options and easy to follow throughout the scenarios. The users would not need to remember how to carry out an instruction.</i></p>
<p><b>7. Flexibility and efficiency of use</b></p> <ul style="list-style-type: none"> <li>• Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user so that the system can cater to both inexperienced and experienced users.</li> <li>• Allow users to tailor frequent actions.</li> </ul>	<p><i>This site shows process steps clearly, so user can just follow the steps for any tasks. Features that are commonly used are in dashboard to easily navigate.</i></p>
<p><b>8. Aesthetic and minimalist design</b></p> <ul style="list-style-type: none"> <li>• Dialogues should not contain information which is irrelevant or rarely needed.</li> <li>• Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.</li> </ul>	<p><i>The design is very clean and straight forward, there are no areas where anything seems to be out of the design theme and it all blends together well.</i></p>
<p><b>9. Help users recognize, diagnose, and recover from</b></p>	<p><i>There were some areas which were overlooked but mainly the edge cases where users could make a mistake were</i></p>



<p><b>errors</b></p> <ul style="list-style-type: none"><li>• Expressed in plain language (no codes)</li><li>• Precisely indicate the problem</li><li>• Constructively suggest a solution.</li></ul>	<p><i>covered well with instructional dialogue boxes which clearly explained the mistake.</i></p>
<p><b>10. Help and documentation</b></p> <ul style="list-style-type: none"><li>• Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation.</li><li>• Help information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.</li></ul>	<p><i>There is some documentation on the consumer end when it comes to SNAP/EBT payment processing which is helpful</i></p>

Appendix II: Heuristics Self-Evaluation

Heuristics Evaluation of Team # 1132 Date 3/18/2021

By Team # 1132 (names) Ashley Kelly, Olivia Kiklica, Bobby Bloomquist, Clark Mahaffey

Part 1: Task Evaluation

<b>Heuristic</b>	<b>Evaluation</b>  As each task is performed, enter in the space below your observations and evaluation of the degree to which the relevant heuristic has been satisfied. Note: Not every heuristic may be relevant for each task. Use as much space as you see fit.				
	<b>Task 1</b>	<b>Task 2</b>	<b>Task 3</b>	<b>Task 4</b>	<b>Task 5</b>
	A producer has just registered for the site, the home dashboard is open, and they want to authorize PayPal as an accepted payment method.	A registered producer has logged into the site, the home dashboard is open, and they want to list a new item for sale.	The registered producer has logged into the site, the home dashboard is open, and they want to view and evaluate any pending sales.	A consumer is looking to find and add produce to their cart. They have the Mobile Market Place open.	A registered consumer has Mobile Market Place open, and they want to checkout and buy the items currently in their cart.

<p><b>1. Visibility of system status</b></p>	<p>The visibility is great, each window has a heading that makes it clear where the user is. The icons are universally understood.</p>	<p>There isn't much of a distinction between the Storefront versus the screen to Add Produce. Maybe a heading/title would help.</p>	<p>Pending sales icon is distinct and eye-catching, so that producers will <i>know</i> if they have a sale pending. The pages thereafter could use bold headers to make them clearer (e.g, 'PENDING SALES.'). Also, there should probably be another page showing the current status of an order (after the sale has been confirmed and before it has been completed).</p>	<p>The highlighted menu icons at the bottom of the app make it very clear which tab the user is currently on. There is no uncertainty where the user is.</p>	<p>There are no loading menus to inform the user that their order is being processed, but cart items are visible. Additionally, there is no message informing the user of their order status after they checkout their cart. Other intermediate statuses are clear, however.</p>
<p><b>2. Match between system and the real world</b></p>	<p>N/A</p>	<p>N/A</p>	<p>N/A</p>	<p>N/A</p>	<p>Fulfilled, cart model mimics real life shopping cart and checkout behavior.</p>
<p><b>3. User control and freedom</b></p>	<p>At every stage there is the menu of tabs on the left side that allow users to navigate wherever whenever.</p>	<p>The cancel buttons and filters allow the user to control their experience.</p>	<p>Back button in upper left corner can easily take user back to the pages where they came from, but there</p>	<p>Overall, the user can navigate back to the main tabs by clicking on the icons on the menu. However,</p>	<p>User cannot delete items from cart or clear cart. No back button from checkout menu so</p>

	Certain menus have a clear X mark or trash can to undo actions.		isn't a 'main menu' icon or an option to navigate to other parts of the website from the pending/accept/decline pages.	there is no explicit cancel button. Once in the cart, there is no easy way to cancel the order. If the user has already searched for an item, there is no clear back button to return to the main list of items.	navigation is not clear.
<b>4. Consistency and standards</b>	The pages are consistently laid out, and the icons are standard. One thing I noticed is that the terms for payment method/payment information/receiving payment are all different.	The screens have a consistent format with the right tab and the left items.	The 'flow' of this task within the application is very clear and aided with visual cues. I don't think there's anything ambiguous here.	The menu icons are very consistent with other web and mobile applications. In addition, when in the cart, the amount of money due is laid out in the same way that a paper receipt is laid out, making it simple to understand. The 'Add to Cart' button and cart interface is	Based on universal cart system in online shopping so consistent and clear to understand.

				also familiar to people who use Amazon.	
<b>5. Error prevention</b>	The prototype accounts for all the errors it can. You can't prevent someone from typing credentials that aren't legitimate.	The errors are prevented by not being allowed to add an item with null/blank fields. Is the USDA option defaulted to either yes or no? Maybe having an 'OR' between them would help.	I think a secondary pop up after you hit 'Accept' or 'Deny' with the follow up, "Are you <i>sure</i> you want to accept/deny this order?" would be good practice. Error prevention is sufficient otherwise.	Slips are avoided in the market search in one way by adding a helpful message if the quantity to buy exceeds the amount the producer has listed.	No option to delete item or empty cart.
<b>6. Recognition rather than recall</b>	The user isn't required to remember information from screen to screen.	Having all the produce options together on the screen helps ease the memory load.	Looks good. Maybe some text or color could be added to the 'accept' and 'decline' buttons. I'm also not sure that you should be able to accept or decline before viewing the order details?	The menu items on the bottom are consistent across all screens, and the 'Add to Cart' button on the market view screen is consistent with the cart icon on the menu, making it clear where the items go when added to the cart.	Satisfied, as checkout and payment are very clear.

<p><b>7. Flexibility and efficiency of use</b></p>	<p>The next time payment information is entered, hopefully you should be able to skip the tutorial.</p>	<p>The filters allow for users to individualize their experience. There is no way to go back to the home screen.</p>	<p>There are no shortcuts to other pages. As mentioned above in the 'user control' section, the interface would benefit from a global menu somewhere along the top.</p>	<p>There is currently no shortcuts or other routes to get from the market to the cart other than the menu icons. Maybe if there was a button that takes the user directly from the market to paying, it would increase the flexibility.</p>	<p>User cannot customize cart or order before checkout.</p>
<p><b>8. Aesthetic and minimalist design</b></p>	<p>The design is very minimalist, I'm impressed :)</p>	<p>The design might be too minimalist because there aren't many indicators whether you're on the Storefront or Add Product page.</p>	<p>Design is consistent and minimalist.</p>	<p>There are no excessive or extraneous icons or text. Every piece is important.</p>	<p>Menus are very simplistic with only key information displayed.</p>
<p><b>9. Help users recognize, diagnose, and recover from errors</b></p>	<p>Although there is an error message when an account is not found, there's not a great way to know what to do next. Do you just</p>	<p>The pop-up when all the fields aren't filled in is a good error indicator. There should also be preventions in place later so they can't enter a negative</p>	<p>Errors exist in their proper place and give enough information to recover from the errors. However, I think there should be a text box that</p>	<p>On search failures and quantity errors, there are helpful error messages that tell the user what went wrong and how to fix it. There may be</p>	<p>Errors with payment methods handled and communicated. Cart and total is shown before checkout.</p>

	exit the pop-up tab?	number and things like that.	appears when you select 'Other' as a reason to decline, and an error will pop up if the textbox isn't filled out.	another error that isn't addressed: on the market screen, the quantity field may not be filled out or have a 0. There should be an error pop-up that helps with this mistake.	
<b>10. Help and documentation</b>	The starting screen tutorial is great.	N/A	N/A	N/A	N/A

Part 2: General Heuristic Evaluation of Prototype Interface

<b>Heuristic</b>	<b><i>Evaluation</i></b>
<p><b>1. Visibility of system status</b></p> <p>Always keep users informed about what is going on.</p> <p>Provide appropriate feedback within reasonable time.</p>	<p>In the space below, enter your observation and evaluation of the degree to which the heuristic has been satisfied. Use as much space as you see fit.</p> <p>On every consumer screen, there is a menu at the bottom with the current tab highlighted. This makes it very clear to the user where they currently are and how to navigate to other tabs. The producer payment tabs are clearly marked, but the producer adding produce tabs are not clearly marked. Error messages handled for all possible errors that could occur while using the application, however, more messages related to successful actions such as checkout could be beneficial.</p>
<p><b>2. Match between system and the real world</b></p> <p>Speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms.</p> <p>Follow real-world conventions, making information appear in a natural and logical order.</p>	<p>On the producer home screen, the information appears in a natural and logical order. However, the home dashboard could be improved by the addition of images within the boxes (the layout of the boxes themselves is fine). The storefront, for example, could feature a close up of a market stall; the sales history an icon of a ledger, etc. These assets will improve visual recognition of features.</p>
<p><b>3. User control and freedom</b></p> <p>Users often choose system functions by mistake.</p> <p>Provide a clearly marked "out" to leave an unwanted state without having to go through an extended dialogue.</p> <p>Support undo and redo.</p>	<p>When a consumer wants to view their cart, they only see a button to proceed to checkout. There is no way to cancel an order or delete an item from the cart.</p> <p>Producer product could use a universal menu (making every page accessible from any place on the website). Back buttons are existent but not consistent – the 'add produce' section could use a cancel button. Also, there is not a logout button anywhere. However, the existing cancel buttons and filters allow for users to have control.</p>



<p><b>4. Consistency and standards</b></p> <p>Users should not have to wonder whether different words, situations, or actions mean the same thing.</p> <p>Follow platform conventions.</p>	<p>The producer interface seems very clear in its aesthetic, functions, and focuses. Nothing seems to cause undue confusion by overlapping.</p> <p>For the consumers, the menu icons are very consistent with other mobile app icons. The total amount of money due is also split up in the same way that a receipt is, making the breakdown of costs very clear to the users. The 'Add to Cart' and Cart interface also stays very consistent with most other online shopping applications, like Amazon.</p>
<p><b>5. Error prevention</b></p> <p>Even better than good error messages is a careful design which prevents a problem from occurring in the first place.</p>	<p>For producers, the use of checkboxes and lists across the platform reduces the likelihood of the user encountering an error (for example, if they had tried to list apples for sale, but the way they had done so was not recognized by the system). We need to be cautious in the future of errors like inputting negative numbers.</p>
<p><b>6. Recognition rather than recall</b></p> <p>Make objects, actions, and options visible.</p> <p>User should not have to remember information from one part of the dialogue to another.</p> <p>Instructions for use of the system should be visible or easily retrievable whenever appropriate.</p> <p>Buildings are all visible and clickable</p> <p>Experienced users can directly check on the buildings they are interested in by clicking on them</p> <p>Recent searches feature is useful for users that use the same buildings regularly</p>	<p>The consumer app has the menu at the bottom which stays consistent through all screens. When adding produce to the cart, the 'Add to Cart' button is the same as the cart icon in the menu, making it very familiar and easy to understand. The producer site screen is wide enough to allow for the user to see many things simultaneously.</p>
<p><b>7. Flexibility and efficiency of use</b></p> <p>Accelerators -- unseen by the novice user -- may often speed up</p>	<p>For consumers, there is only one way to navigate from the marketplace to the cart, and that is through the menu. There could possibly be a button that adds a given produce to the cart</p>

<p>the interaction for the expert user so that the system can cater to both inexperienced and experienced users.</p> <p>Allow users to tailor frequent actions.</p>	<p>and takes the user to the cart directly. This would increase flexibility and user control. For the producer add produce page, there is no way to return home.</p>
<p><b>8. Aesthetic and minimalist design</b></p> <p>Dialogues should not contain information which is irrelevant or rarely needed.</p> <p>Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.</p>	<p>The consumer screens are very clear and minimalist. There are no extraneous images, icons, or text. The producer site was also minimalist, but we need to remember to label things properly.</p>
<p><b>9. Help users recognize, diagnose, and recover from errors</b></p> <p>Expressed in plain language (no codes)</p> <p>Precisely indicate the problem</p> <p>Constructively suggest a solution.</p>	<p>There are various error messages for consumers. When adding items to the cart from the marketplace, if the quantity is too high, there is a message. However, there is no error pop-up if the quantity is 0 or empty, which will be a problem since users will not be able to quickly diagnose their issue. The error messages for consumers are good, we just need to remember to implement specific ones like inputting negative numbers.</p>
<p><b>10. Help and documentation</b></p> <p>Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation.</p> <p>Help information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.</p>	<p>There is not currently any help pages for either the consumer or producer documents. This will need to be implemented, at the very least as a FAQ page.</p>

*Appendix III: Heuristics Evaluation by Course Professors*

**Notes Taken During the Evaluation of Team # 1132 Date 3/22/2021 By  
Course Professors**

Scenario 1:

- PayPal – capitalize
- Error statements should have no exclamation points
- Add a warning box when the user clicks on delete current Payment method.

Scenario 2:

- Say apple/banana instead of Product A/B, etc.
- Like the shading and coloring
- Feedback message that says “Produce C added” would be helpful.

Scenario 3:

- Honey crisp apple is misspelled
- Other option – should be able to fill in a reason why
- Invalid address – maybe out of delivery area
  - Software can handle this – geofencing
- Add a text box under “Other” in the “Deny” page. On the side, how would the seller know if an item is out of stock? Is there a tracking system that helps them to do that?
- Add a warning box when user clicks “Deny” or “Confirm”.

Scenario 4:

- Is intent a static pull of quantity? Or only something that gets checked at cart?
  - Multiple people buying at same time
  - Customers like static pull according to Ronnie
    - Better to know they can’t buy something now vs. later
- Search term is apple, but no apple in the names
  - Search based off keyword or categories?
- Add a small notification when the user clicks “add to cart” so they know that it’s in their cart.
- Add removing items in the cart. Add the warning box when removing.
- If quantity entered is zero, there should be a dialogue box correcting the user.

Scenario 5:

- None