TILITER

Recognition API
UI/UX Best Practice Guide
For Checkouts

Table of Contents

1. Introduction

The purpose of this guide is to assist developers and UI/UX designers in building an intuitive User Interface to the Tiliter Recognition API in their new or existing checkout software.

It should be followed along with the API documentation during the planning, design, and development phases of integrating with the Tiliter Recognition API.

It is important to note that the User Interface can have a dramatic effect on the performance of the product, including both user satisfaction and cost savings.

2. Scope

This document provides best practice guidelines on the end-to-end User Experience of the Tiliter Recognition API as integrated into a Point-of-Sale (POS) system, such as an Assisted Checkout or a Self-Serve Checkout.

Best practice UI/UX integration will be described, covering every touch point of the experience.

Example screenshots are provided, and these can be used as references for integrating the Tiliter Recognition API into a new or existing POS environment.

This document covers the following aspects of integration:

- Initiating a recognition event – what it looks like to the user
- Presenting selections to the user, including the product and bag state (bag or no bag)
- Adding products to a cart
- Handling error states

The intended audience of this document are Product Designers, UI Designers, UX Designers, Product Managers, Business Analysts, Software Developers, and anyone working on a project to integrate the Tiliter Recognition API into a new existing POS system.

Equally important, but not covered in this document, are the technical aspects of integration:

- Uploading an image and requesting a recognition event from the Tiliter Recognition API
- Interpreting a response from the Tiliter Recognition API
- Sending a user selection back to the Tiliter Recognition API
- Receiving a fraud flag from the Tiliter Recognition API
- Sending a completed transaction event to the Tiliter Recognition API

These are covered in the Tiliter Recognition API Specification Document, which can be found here: https://doc.recognition.services.tiliter.com/

3. Conventional Checkout Workflow

First, an overview of a typical conventional checkout workflow is explored. This will allow us to understand the conventional checkout user experience, which will help frame the intended Tiliter Recognition enhanced user experience.

There are two primary users of a checkout: the cashier (using an Assisted Checkout) and the shopper (using a Self-Serve Checkout). Both will be described below.

3.1. Cashier (Assisted Checkout)

Case 1: Entering product ID

1. The cashier picks up a product without a barcode and places it on the scale
2. The cashier taps on the product ID input field
3. The cashier enters the product ID which they have either memorised or read from some reference material, and presses enter
4. The POS adds the product along with its weight and price information to the virtual cart
5. If a mistake is made, the cashier can remove it from the virtual cart directly

Case 2: Searching through product select screen

1. The cashier picks up a product without a barcode and places it on the scale
2. The cashier taps on the product select button
3. The cashier navigates through the product search menu to find the correct product and taps on it
4. The POS adds the product along with its weight and price information to the virtual cart
5. If a mistake is made, the cashier can remove it from the virtual cart directly

3.2. Shopper (Self-Serve Checkout)

1. The shopper picks up a product without a barcode and places it on the scale
2. The shopper taps on the produce select button
3. The shopper navigates through the produce search menu to find the correct produce and taps on it
4. The POS adds the product along with its weight and price information to the virtual cart
5. If a mistake is made, they must ask for assistance to reverse it

4. Design Principles

For the software to work as intended, it must be integrated in a way that is well designed and seamless. Any friction points or barriers, no matter how small, will result in lower adoption and the efficiency and accuracy gains from the Tiliter Recognition API will not be realised.

The following design principles can be used to guide the User Interface design:

1. **Display products prominently**: Predicted products must be displayed to the user prominently and conveniently without any barriers to select them, so that users intuitively select from the shortlist without explicit instructions or training

2. **Use intuitive graphics**: Most people respond well to strong visual cues. Leverage this and accompany products with large photos of the product, which are clear, vibrant, and in the correct aspect ratio. This helps the user select the correct product quickly.

3. **Give the user control**: Never force the user to do something they don't want to do. Always allow experienced users who wish to use conventional means to select products do so without any additional barriers. These users will be converted after they see the predictions are correct enough times.

4. **Avoid disruptions to flow**: Respect the shopper's time. Do not show long or complex messages that won't be read or hide things in buttons that won't be found. Don't interrupt the flow, even if there is a failure. Instead, revert to the conventional product selection method.

5. **Confirm before committing**: If an action cannot be undone without help from an assistant, then do not do it automatically. Instead, always require the user to tap a button before performing the action.

6. **Consistency is key**: Keep control and navigation buttons in consistent locations. For instance, if there is a back or search button, it should always appear in the same position on the screen so users can easily familiarise themselves with the interface.

These design principles will be used throughout this document as the entire user journey is explored. They will be put into practice in example designs to demonstrate them in action.

For the initial integration, it is strongly encouraged that these are all followed.

5. Recognition API Workflow

5.1. Overview

When a checkout has been upgraded to use the Tiliter Recognition API, the user workflow changes to the following:

1. The shopper picks up a product without a barcode and places it on the scale
2. The scale stabilises and triggers a request from the Tiliter Recognition API.
3. The product is automatically added to the virtual cart, or shortlist of products predicted are shown on screen
4. If a shortlist is shown, the shopper selects one of the products on screen and that product is added to the virtual cart

This overview is used as the basis of the following sections of this document.

First, several screens will be introduced. These screens are either completely new, or modifications of existing/traditional screens used in the checkout UI.

5.2. Screens

5.2.1. Home Screen

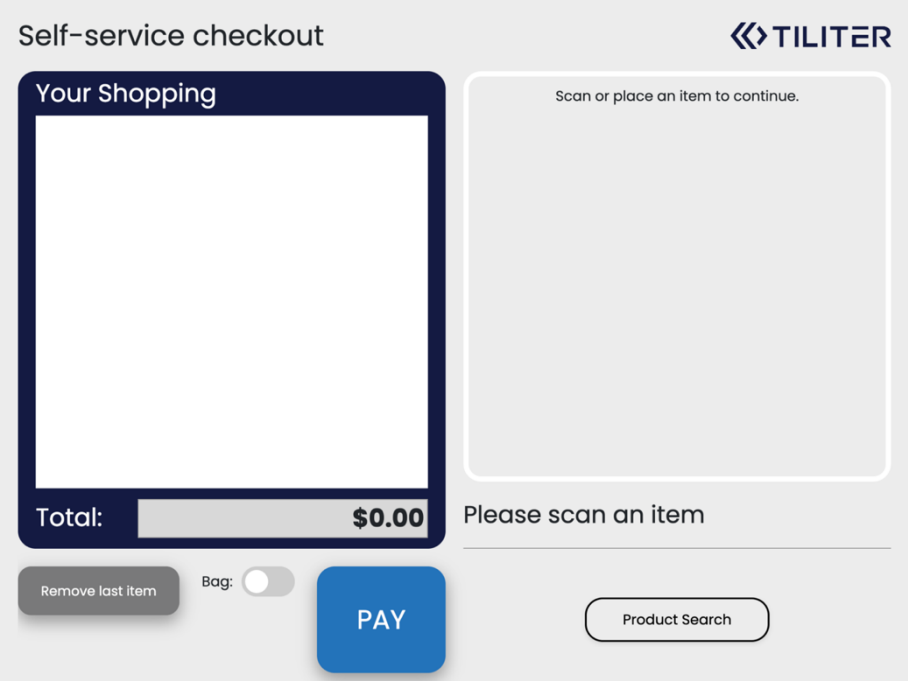

Figure 1 – Example home screen with Tiliter Recognition API

Description	This is an example Home Screen that has been enhanced with Tiliter Recognition. This is what's shown when the system is waiting to be used. It has all the elements that shoppers are familiar with – a virtual cart, a product search button, and the option to finish the transaction and pay. The difference only becomes apparent after a product without a barcode is placed on the scale. At this point, the Recognition API is called, and the results are shown in the Recognition Panel, described in section 5.2.2.
When to show	When the system is waiting to be used, and between scans
Important details	• The home screen looks very similar to a home screen that isn't upgraded with Tiliter Recognition • The Recognition Panel uses the same design elements as the remainder of the Home Screen. • When a product is placed on the scale, it automatically triggers a recognition event and brings up a prediction in the Recognition Panel, shown in sections 5.2.2, 5.2.2.3, and 5.2.2.4 • Traditional checkout workflow is still available - barcoded items still scan and add to the virtual cart, and a Product Search button is available to use

5.2.2. Recognition Panel

5.2.2.1. Overview

The Recognition Panel is a section on the home screen that is reserved to present results from the Tiliter Recognition API. It will be shown in context in the following sections.

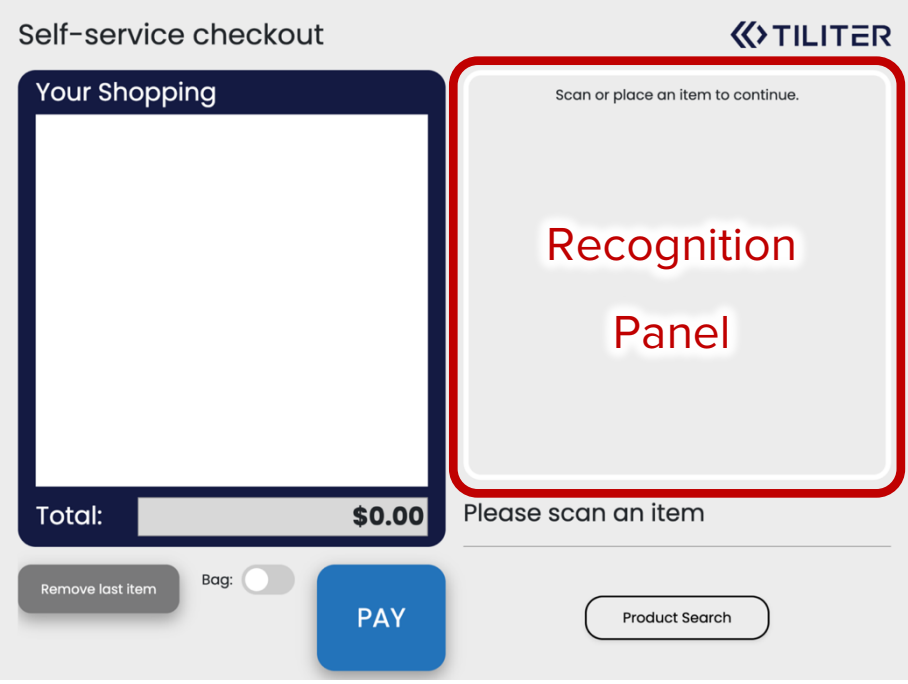

Figure 2 – The Recognition Panel highlighted on the home screen

This panel allows the user to interact with the output of the Tiliter Recognition API, which sends product codes in order of confidence, i.e. the first result in the list is the most confident, followed by the second, third etc (if they exist). See the *Recognise Product* section in the Tiliter Recognition API specification for more details: https://doc.recognition.services.tiliter.com/

This order must be honoured in the Recognition Panel. The order should be left to right, then top to bottom, as follows:

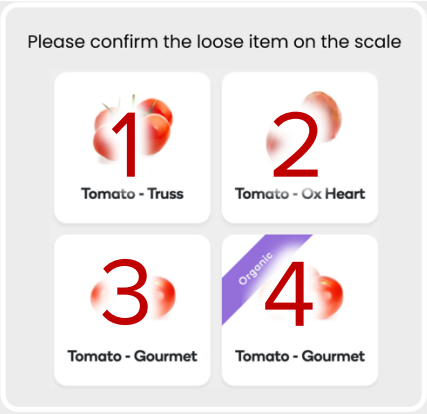

Figure 3 - The order products are presented in the Recognition Panel

5.2.2.2. One Selection

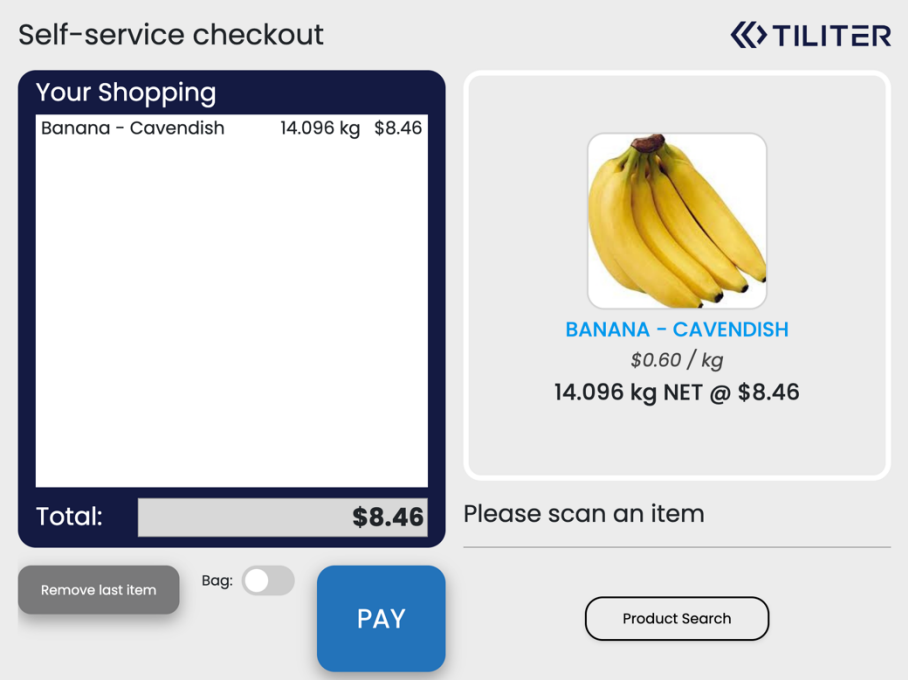

Figure 4 – A recognition event with a single product automatically added to the cart

Description	Here, Tiliter Recognition API only predicted a single item. This typically happens when it is quite confident. The product was added to the virtual cart since it was a single prediction only.
When to show	After the following sequence of events occurs: 1. The weight on the scale stabilises 2. The POS sends a prediction request to the Tiliter Recognition API 3. The Tiliter Recognition API responds with one prediction
Important details	• The product is added to the cart just like any other product would be added if it was searched for manually, or scanned in via a barcode • The product has a clear photo that has the correct aspect ratio • Since the product was automatically added to the cart, a button is provided to remove it from the cart ○ If it is not the store's policy to allow this, a confirmation message to add the product to the cart should be shown first.

5.2.2.3. Two selections

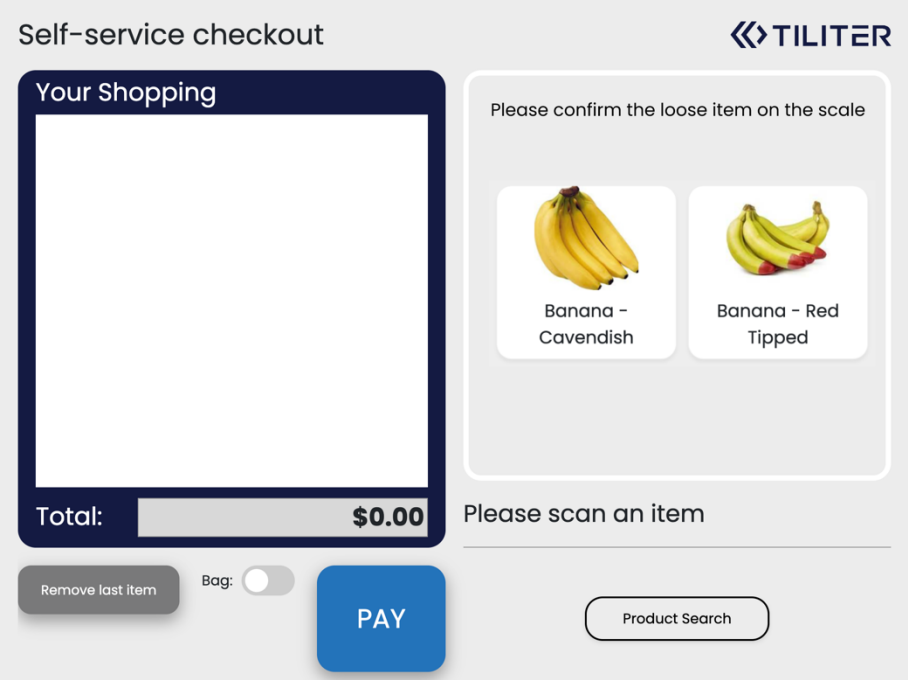

Figure 5 – Prediction screen showing two selections

Description	Here, the user is presented with a shortlist of produce that was predicted by the Tiliter Recognition API. In this case, two products were predicted, and so two are shown in the Recognition Panel, along with a photo of each.
When to show	After the following sequence of events occurs: 1. The weight on the scale stabilises 2. The POS sends a prediction request to the Tiliter Recognition API 3. The Tiliter Recognition API responds with two predictions
Important details	• Since the Tiliter Recognition API responded with two products, two products are shown to the user • The products have a clear photo that has the correct aspect ratio • The product search button is available for the shopper to select something else if the correct product is not shown • The product on the left was the first product in the ordered list provided by the Tiliter Recognition API

5.2.2.4. Four selections

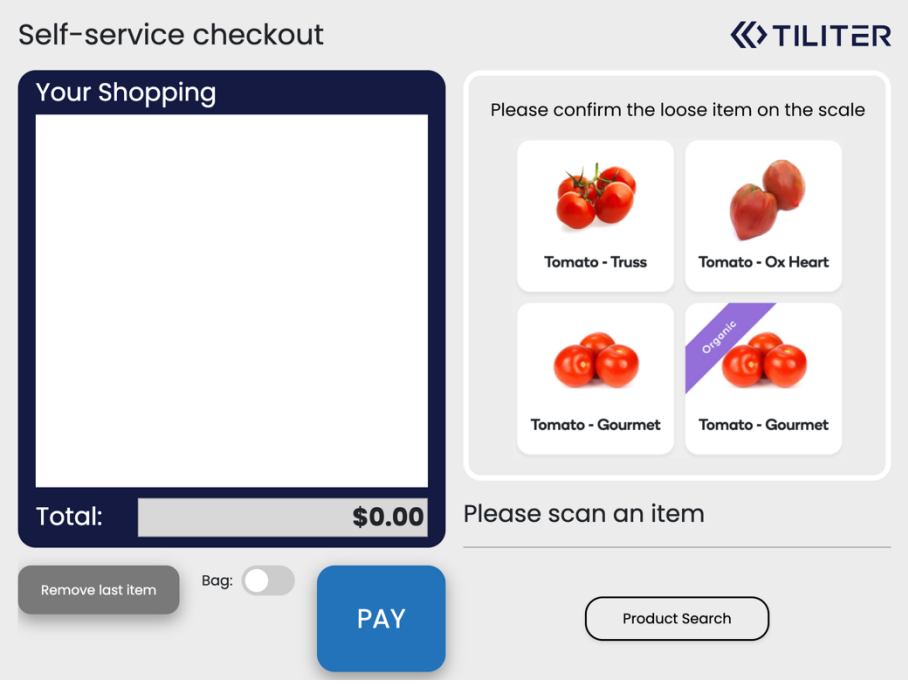

Figure 6 – Prediction screen with four results

Description	Here, the user is presented with a shortlist of produce that was predicted by the Tiliter Recognition API. In this case, four products were predicted, and so four are shown in the Recognition Panel, along with photos of each.
When to show	After the following sequence of events occurs: 1. The weight on the scale stabilises 2. The POS sends a prediction request to the Tiliter Recognition API 3. The Tiliter Recognition API responds with four predictions
Important details	• Since the Tiliter Recognition API responded with four products, four products are shown to the user • The products have a clear photo that has the correct aspect ratio • The product search button is available for the shopper to select something else if the correct product is not shown • The product on the left was the first product in the ordered list provided by the Tiliter Recognition API, followed by the top right, bottom left and then bottom right

5.2.3. Quantity confirmation

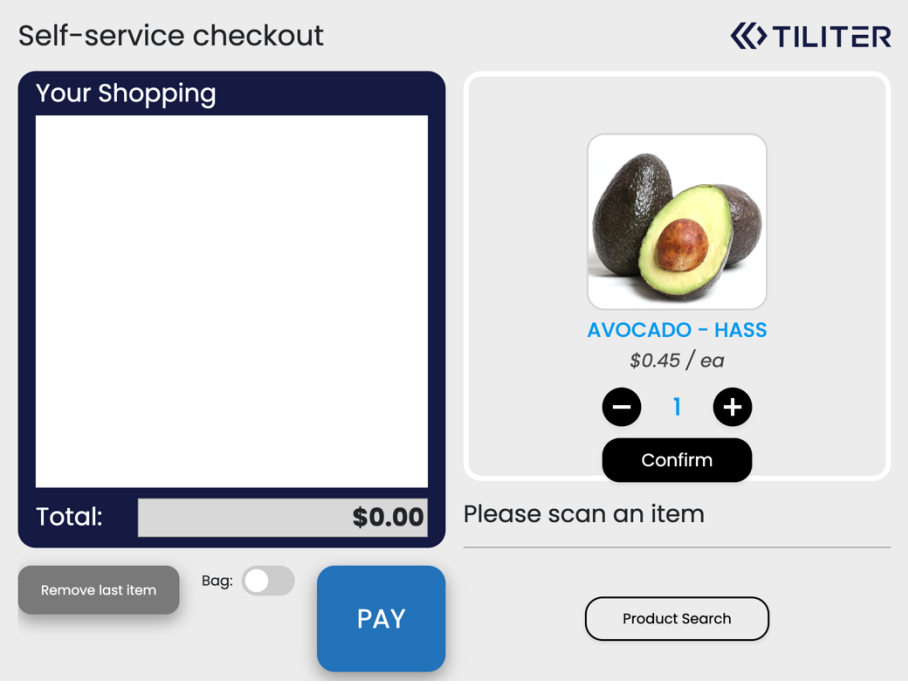

Figure 7 – Quantity confirmation screen

Description	This screen is used when a product that is sold by quantity has been selected using the Recognition panel, or manually via the Product Search button. The screen allows the user to select the quantity required, and it is pre-filled with a quantity estimate based on the weight and average weight of the product.
When to show	After the following sequence of events occurs: 1. The weight on the scale stabilises 2. The POS sends a prediction request to the Tiliter Recognition API 3. Either of the following: a. The Tiliter Recognition API responds with one product (sold by quantity), which is automatically added to the virtual cart b. The Tiliter Recognition API responds with two or more products, and the user selects one of them (sold by quantity), which is added to the virtual cart c. The user manually searches for, and selects a product (sold by quantity), which is added to the cart
Important details	• The quantity count is automatically populated using the Recognition API details • The quantity count can be adjusted, and confirmed • If the confirmation message is accepted, the item is added to the virtual cart • If the quantity is adjusted before confirmation, the quantity to be added changes

5.2.4. Adding to cart

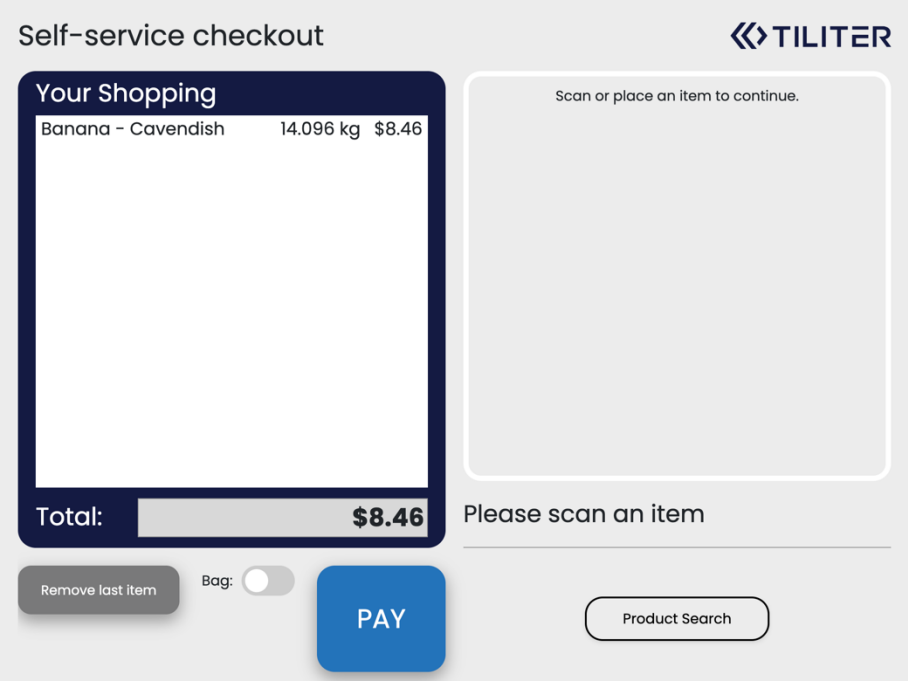

Figure 8 – A product sold by weight in the virtual cart, after it was picked up from the scale, readying the system for another product

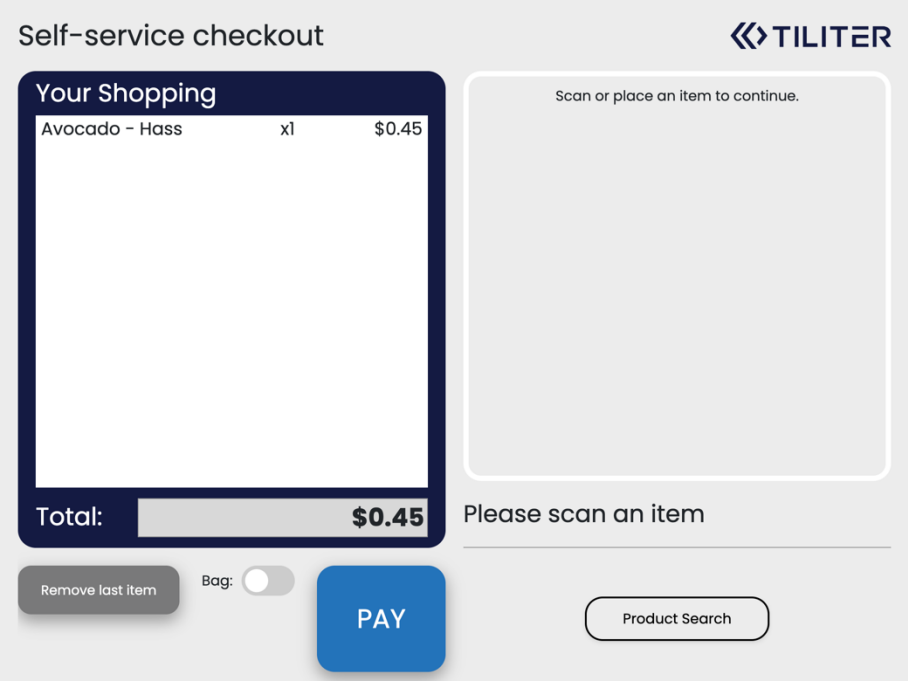

Figure 9 – A product sold by quantity in the virtual cart, after it was picked up from the scale, readying the system for another product

Description	These screens show the virtual cart with an additional product in them, after the user has been through either of the workflows presented in sections 5.2.2.2, 5.2.2.3, and 5.2.2.4, and the product is picked up from the scale. This should look the same as if the user manually searches for and selects the product.
When to show	After the following sequence of events occurs: 1. The weight on the scale stabilises 2. The POS sends a prediction request to the Tiliter Recognition API 3. Either of the following: a. The Tiliter Recognition API responds with one product (sold by weight), which is automatically added to the virtual cart b. The Tiliter Recognition API responds with two or more products, and the user selects one of them (sold by weight), which is added to the virtual cart c. The user manually searches for, and selects a product (sold by weight), which is added to the cart 4. The product is removed from the scale
Important details	• The product along with its details are shown in the virtual cart • Products sold by weight have their weight and total price • Products sold by quantity have their quantity and total price

5.3. Bringing it all together

With all the relevant screens introduced, several examples will be explored to show how it all works together. Note, this is not intended to be an exhaustive list of all possible user workflows. Instead, illustrative examples are provided and combinations of these are expected to work in a similar fashion.

5.3.1. Example 1 – Confident recognition

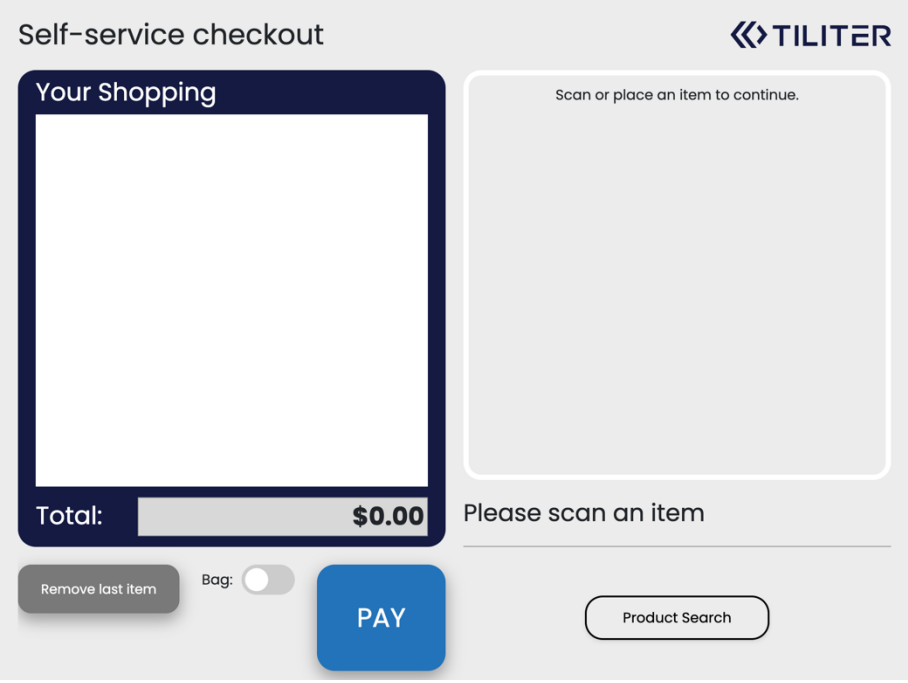

Figure 10 – The home screen

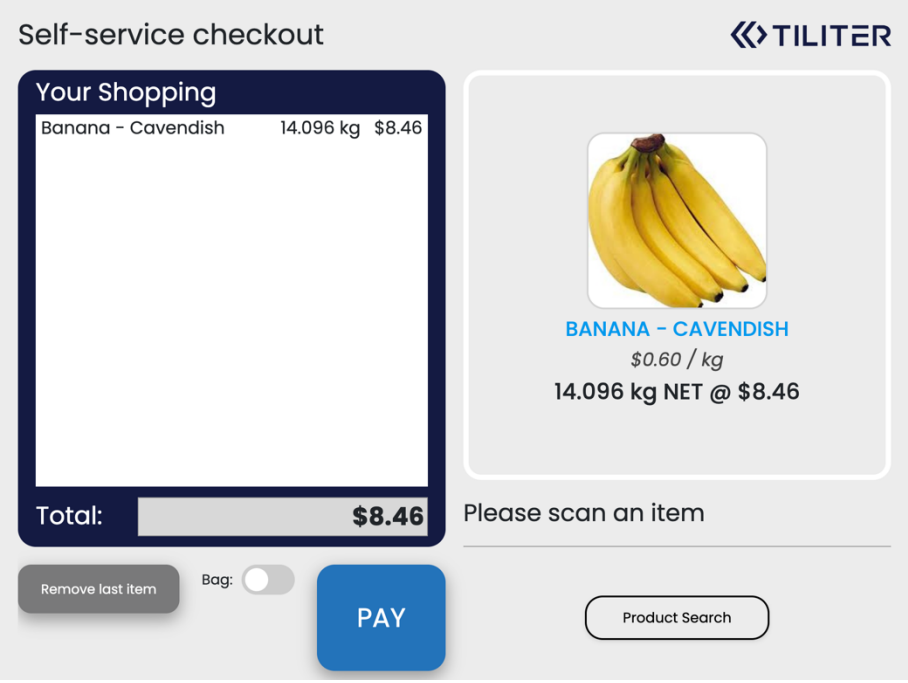

Figure 11 – After the bananas are placed on the scale, it is automatically recognised and added to the virtual cart

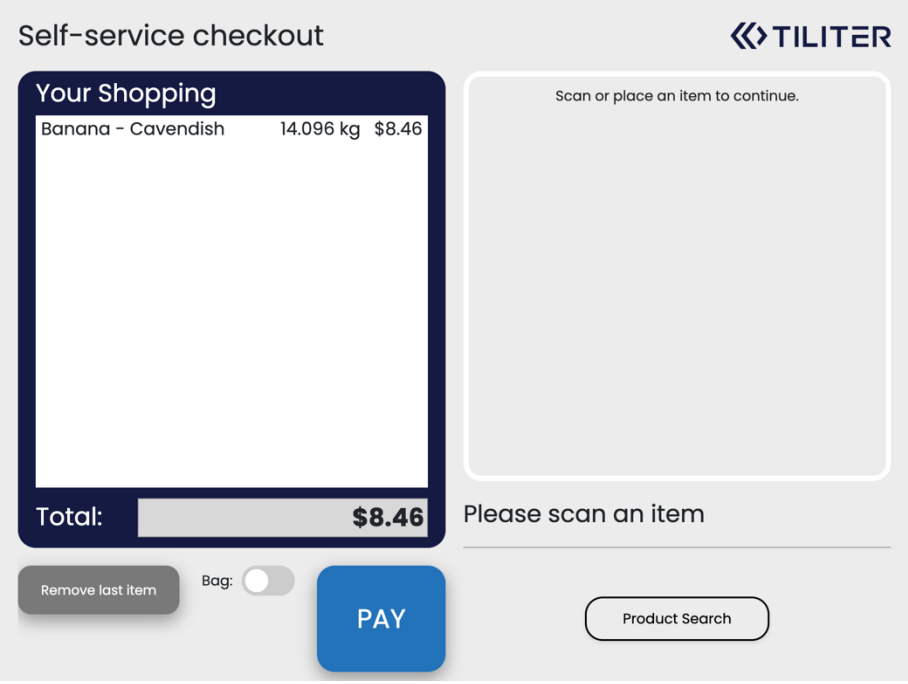

Figure 12 – After the bananas are picked up from the scale, the system is ready for another product

5.3.2. Example 2 – Selection between two products

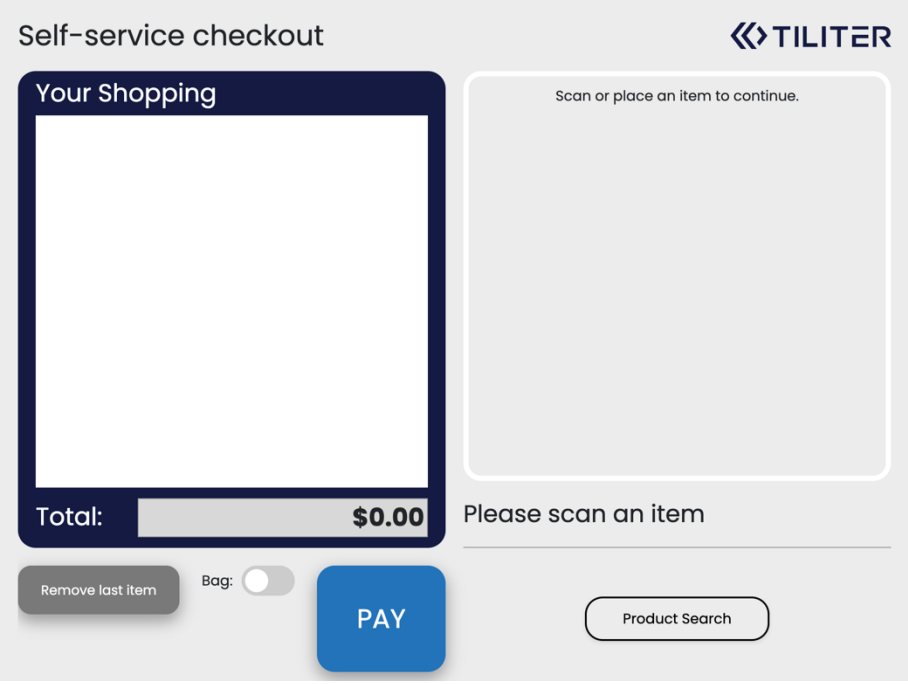

Figure 13 – The home screen

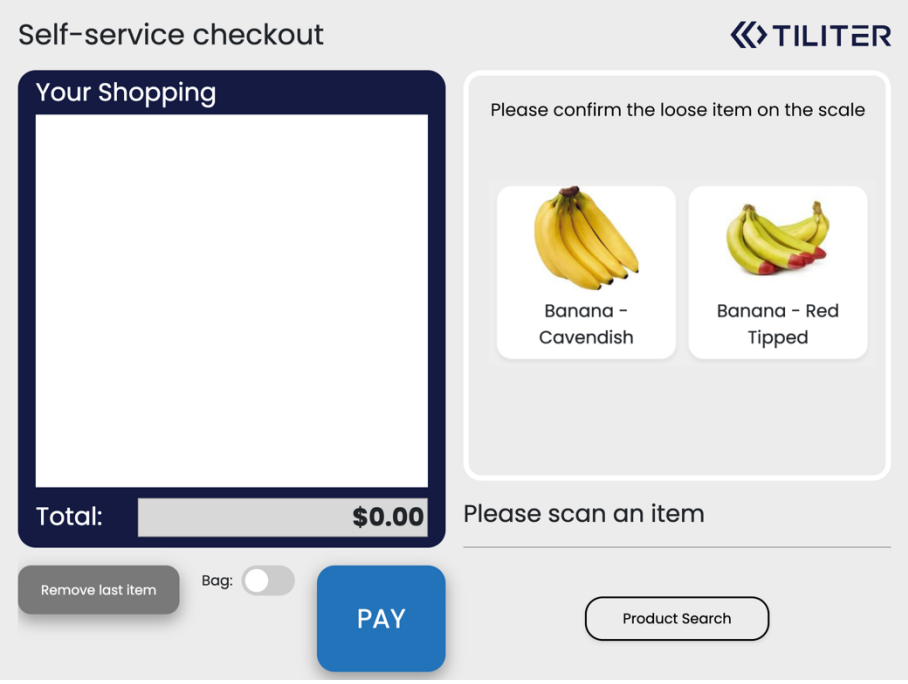

Figure 14 – A shortlist of two choices is provided to the user

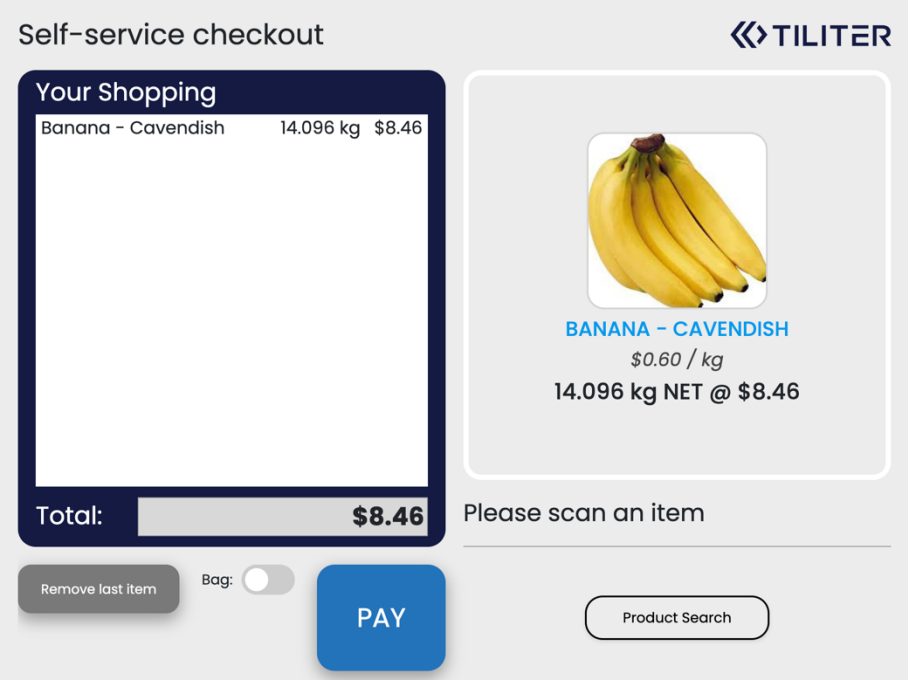

Figure 15 – After the user selects a choice, it is added to the virtual cart.

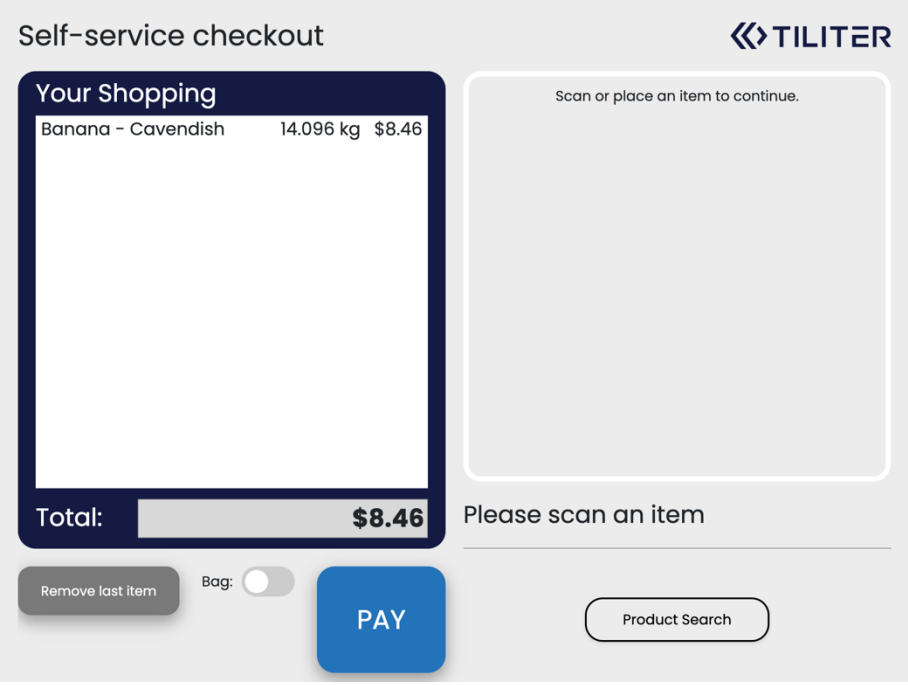

Figure 16 – After the bananas are picked up from the scale, the system is ready for another product

5.3.3. Example 3 – Selecting a product sold by quantity

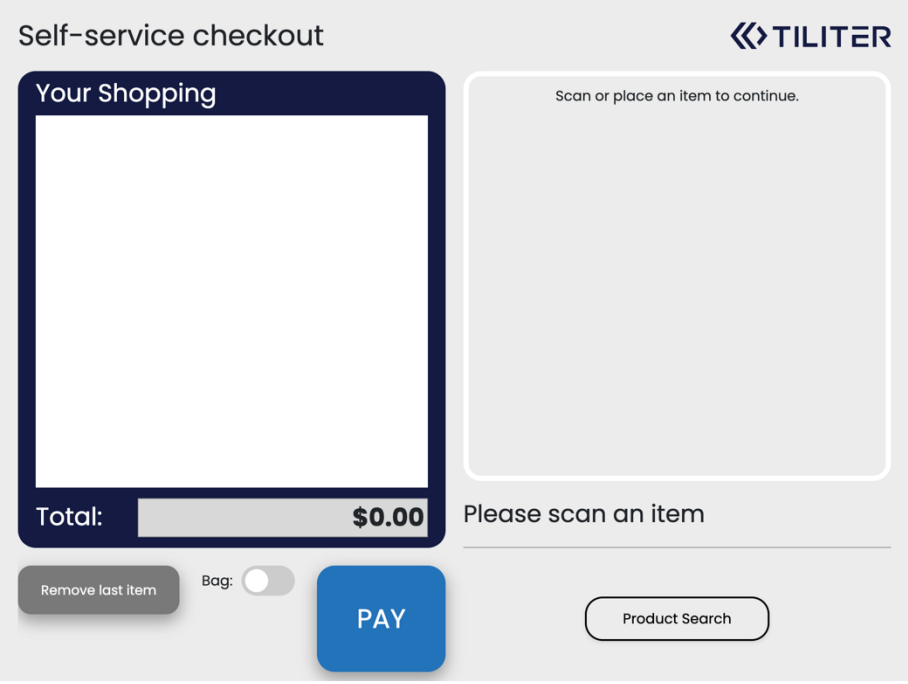

Figure 17 – The home screen

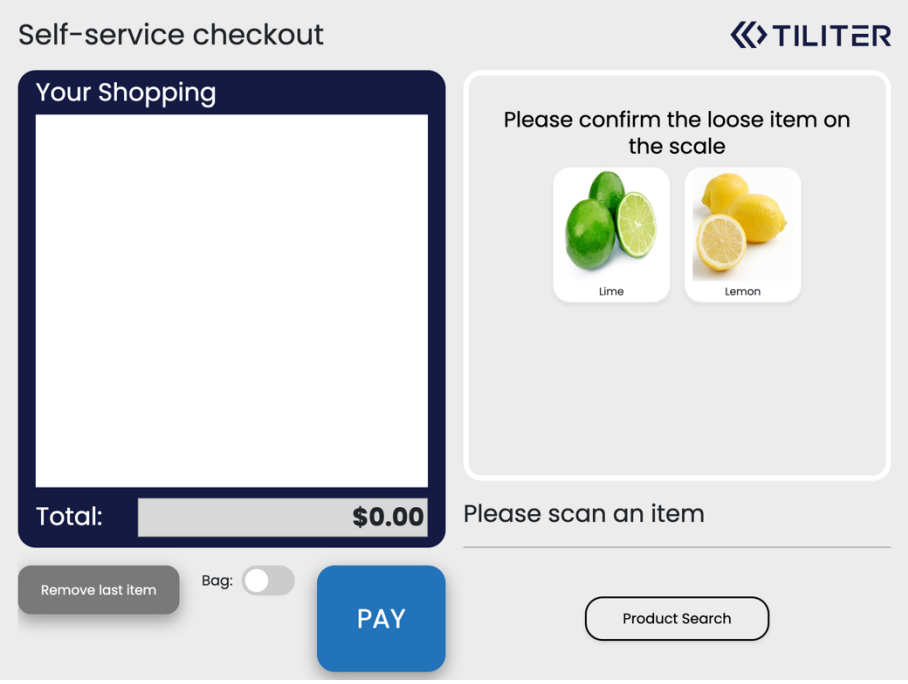

Figure 18 – A shortlist of two choices is provided to the user

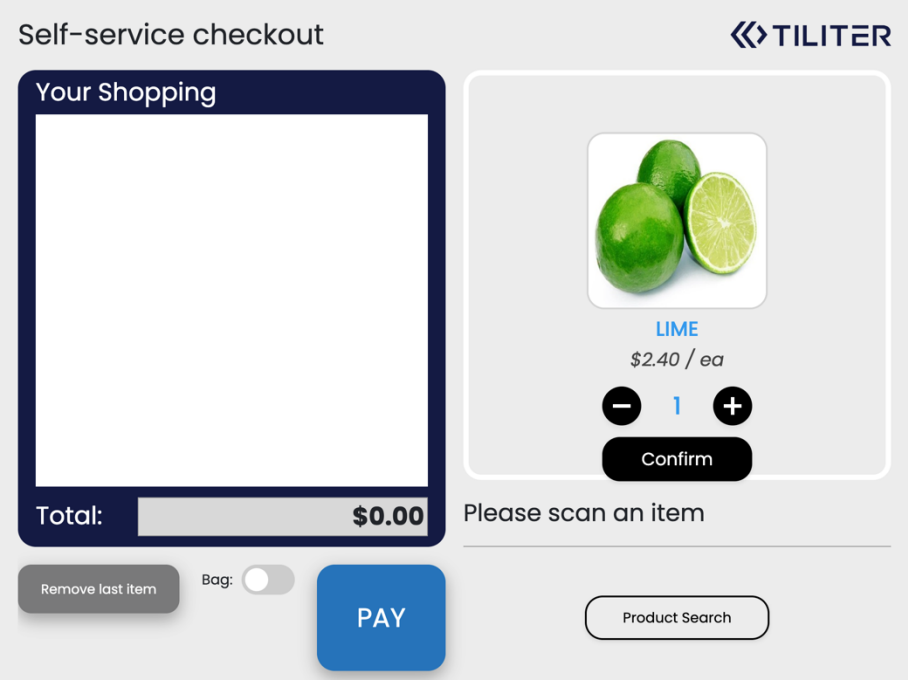

Figure 19 – After a product that is sold by quantity is selected, a quantity confirmation screen is shown

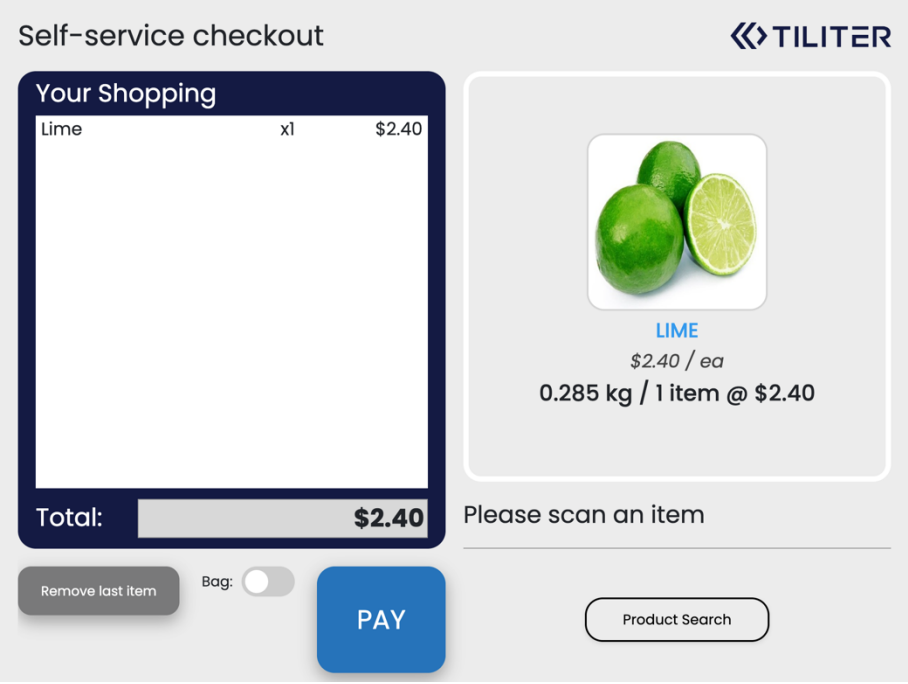

Figure 20 – After the quantity is confirmed, it is added to the virtual cart

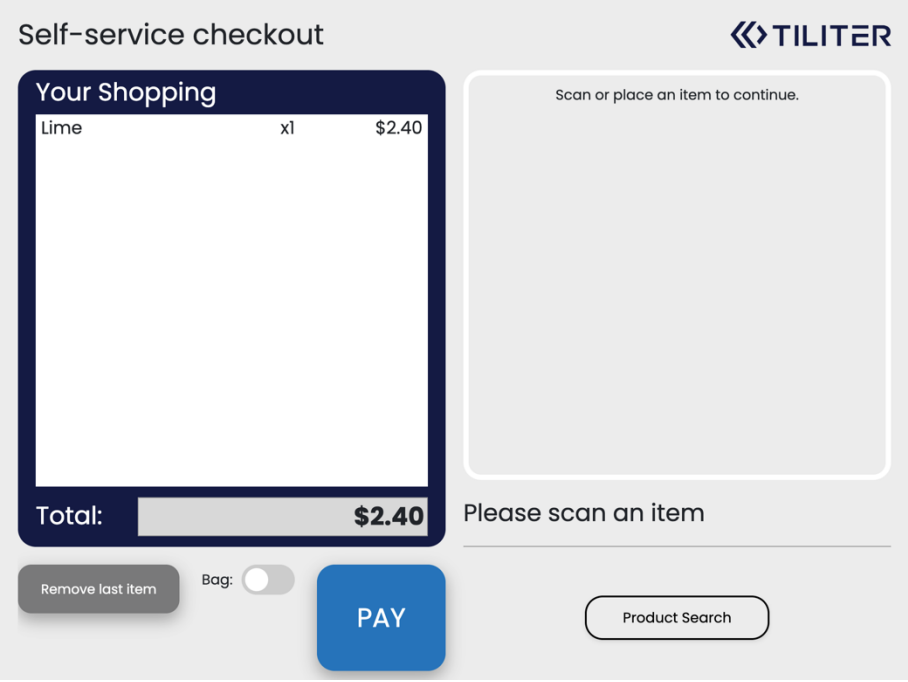

Figure 21 – After the bananas are picked up from the scale, the system is ready for another product

6. A Short Note on Errors

If any part of the end-to-end Recognition system including camera, camera driver, system software, and the Tiliter Recognition API entering any kind of error state, including but not limited to:

- Cannot predict due to bag blocking camera
- Camera is disconnected or has malfunctioned
- Processor has disconnected or has malfunctioned
- Cables of any kind disconnected or damaged
- Network is disconnected or otherwise down
- General software fault

The user should not be presented with an error. Instead, the checkout should function like it would without the Tiliter Recognition API. The basis of this approach means that the user is always able to make a purchase in a streamlined manner without any disruptions. The user is *never* placed in position where they are taken away to a different error screen that they must go back from, and the user journey is not interrupted – which would violate **Design Principle 4 - Avoid disruptions to flow.**

7. User Interface Do's and Don'ts

This section expands upon the Design Principles in section 0 with more specific advice related to the actual User Experience workflow.

Do:

- Use large and clearly legible font for the prediction selection buttons
- Use large, bright, and clear images for the prediction selection buttons
- Use images with the correct aspect ratio for the prediction selection buttons
- After a prediction has been selected, grey it out and disable the button to reduce errors such as double tapping

Don't:

- Show an error if the Tiliter Recognition API system doesn't respond, responds with an error, or contains no predictions. This would violate **Design Principle 4 - Avoid disruptions to flow.**
    - Instead, the user can use conventional means to select a product.
- Automatically add an item to the cart if it's difficult to remove. This would violate **Design Principle 5 – Confirm before committing**
    - Instead, require the user to confirm the produce by tapping the screen. Alternatively, make it easy to remove the item from the cart.
- Allow the user to add something to the cart after it has already been added. This should only be possible after either 1. A new prediction has been made after a stable weight or 2. The item was removed from the cart.
    - Instead, in these cases the buttons should grey out to prevent adding them again

8. Revision History

Revision	Date	Comments
1	16/Feb/2023	Initial release
2	17/Feb/2023	• Fixed broken references • Added description of automatic counting to quantity pages • Converted to portrait
3	24/Feb/2023	• Made minor formatting improvements throughout the document • Updated the selection screen reference screenshots, and their corresponding text • Added section 5.3, which contains examples