# Table5 Kiosk ordering system

## [30199836@my.nclan.ac.uk](mailto:30199836@my.nclan.ac.uk)

Introduction

The client has requested a prototype for a restaurant fast-track kiosk ordering system. The customer should be able to approach the kiosk and start a new order. The menu items should be clearly visible, and the user interface should be intuitive enough that anybody can use it without prior instructions. In the prototype there should be a discount of 10% to entice customers to use the kiosk over waiting in line. When the customer places the order, it would be transferred to the checkout and the customer given a receipt with order reference to collect their order when it’s ready.

User Stories

As a customer, I want to be able to quickly place an order.

As a customer, I want to add multiple items to my order.

As a customer, I want to be able to remove items from my order.

As a customer, I want to easily navigate the menu.

As a customer, I want to see pictures and descriptions of the menu items.

As a customer, I want to clearly see the pricing at every step.

User Interface Design

A screenshot of a menu

Description automatically generated

When the customer clicks a menu item, the add item dialog is shown to allow them to customise the selection before adding the item to their order. Clicking the item in the order will allow the customer to edit or remove the item from the order. The customer can also choose to checkout or cancel the order process.

A screenshot of a menu

Description automatically generated

Pseudocode and Flow charts

**Logic for the main interface**

**A screenshot of a computer screen

Description automatically generated**

**Pseudocode for the elfHash() method**

A quick reliable hashing function is needed to allow fast lookup of the lists. The algorithm below is recommended because of its speed and simplicity.

**SUBROUTINE** elfHash(str)  
 **SET** res = 0

**SET** bytes = **GET** str as a byte array

**FOR EACH** byte in bytes **DO**

**SET** res = rotateLeft(res,4) + byte

**SET** x = res **AND** 0xF0000000

**IF** (x!=0)

**SET** res = res **OR** rotateRight(x, 24)

**SET** res = res **AND NOT** x

**RETURN** res **AND** 0x7FFFFFFF

**Pseudocode for the updateOrderPrice() method**

**SET** total = 0

**FOR EACH** order item **DO**

**SET** total = total + order item price

**SET** subtotal label text to total

**SET** discount = total \* 10%

**SET** discount label text to total

**SET** total label text to (total – discount)

Testing

**Intro Overlay**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Scenario** | **Expected Result** | **Actual Result** | **Comment** | **Pass/Fail** |
| **Normal** | | | | | |
| N1 | Check if videos load and play correctly | Videos load and play and loop correctly | As expected |  | Pass |
| N2 | Progress bar increases as menu is loaded | Progress bar increases | As expected |  | Pass |
| N3 | Start button shows after menu is loaded | Start button appears and progress bar disappears | As expected |  | Pass |
| N4 | Click the about button | About dialog is shown | As expected |  | Pass |
| N5 | Start button clicked | Intro overlay hidden and main menu shown | As expected |  |  |
|  |  |  |  |  |  |

**About Dialog Overlay**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Scenario** | **Expected Result** | **Actual Result** | **Comment** | **Pass/Fail** |
| **Normal** | | | | | |
| N1 | Close button clicked | Dialog should close and the intro overlay appears again | As expected |  | Pass |
| N2 | Click anywhere outside the dialog | Dialog should close and the intro overlay appears again | As expected |  | Pass |
| N3 | Correct version number shown | The correct version number should be shown | As expected |  | Pass |
|  |  |  |  |  |  |

**Main Interface**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Scenario** | **Expected Result** | **Actual Result** | **Comment** | **Pass/Fail** |
| **Normal** | | | | | |
| N1 | Category item clicked | Menu list should scroll to the selected category | As expected |  | Pass |
| N2 | Scroll down and up through multiple categories on the menu list | The category list selection should be updated to show the current category the menu list is scrolled to | As expected |  | Pass |
| N3 | Cancel button clicked | Intro overlay shown and menu interface hidden | As Expected |  | Pass |
| N4 | Menu Item clicked | Add item dialog shown | As Expected |  | Pass |
| N5 | Menu product image hover over | Magnify icon shows on mouse enter and hides on mouse exit | As expected |  | Pass |
| N6 | Menu product image clicked | Image dialog shown with large product image | As expected |  | Pass |
| N7 | Order item clicked | Edit order dialog shown | As expected |  | Pass |
| N8 | Checkout button clicked when nothing in order list | Order Price labels pulsate to indicate that there’s nothing in the order | As expected |  |  |
| N9 | Checkout button Clicked with items in the order | Checkout dialog shown | As Expected |  | Pass |
|  |  |  |  |  |  |

**Add item dialog**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Scenario** | **Expected Result** | **Actual Result** | **Comment** | **Pass/Fail** |
| **Normal** | | | | | |
| N1 | Close button clicked | Dialog hidden and nothing added to order | As expected |  | Pass |
| N2 | Click anywhere outside the dialog | Dialog hidden and nothing added to order | As expected |  | Pass |
| N3 | Correct product details shown from the item clicked on the menu | The product details should match what was selected in the menu | As expected |  |  |
| N4 | Click plus and minus buttons | The quantity increases and decreases within the defined range and the price updated | As Expected |  | Pass |
| N5 | Where available, options and choice sections are shown correctly | The correct choices are shown for each option selected | As Expected |  | Pass |
| N6 | Required choices are shown correctly | Each required choice is labelled as required | As expected |  | Pass |
| N7 | Number of required choices is shown until all requirements are satisfied | “Add to order” is only shown when all requirements are satisfied | As expected |  | Pass |
| N8 | Click the “Add to Order” button with requirements outstanding | The dialog should scroll to the first required choice | As expected |  | Pass |
| N9 | Click add to order with no outstanding requirements | The dialog should close, and the item added to the order list | As expected |  | Pass |
| N10 |  |  |  |  |  |

**Edit order item dialog**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Scenario** | **Expected Result** | **Actual Result** | **Comment** | **Pass/Fail** |
| **Normal** | | | | | |
| N1 | Close button clicked | Dialog hidden and nothing changed in the order | As expected |  | Pass |
| N2 | Click anywhere outside the dialog | Dialog hidden and nothing changed in the order | As expected |  | Pass |
| N3 | Correct product details shown from the item clicked on the order list | The product details should match what was selected in the order list | As expected |  |  |
| N4 | Click plus and minus buttons | The quantity increases and decreases within the defined range and the price updated | As Expected |  | Pass |
| N5 | Click the remove/bin button | The quantity is set to zero | As expected |  | Pass |
| N6 | Update order button clicked with quantity set to zero | The item is removed from the order | As expected |  | Pass |
| N7 | Update order button clicked with any quantity over zero | The quantity of the item is updated in the order and the total order price updated | As expected |  | Pass |
|  |  |  |  |  |  |

**Checkout dialog**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Scenario** | **Expected Result** | **Actual Result** | **Comment** | **Pass/Fail** |
| **Normal** | | | | | |
| N1 | Close button clicked | Dialog hidden and back to the menu | As expected |  | Pass |
| N2 | Click anywhere outside the dialog | Dialog hidden and back to the menu | As expected |  | Pass |
| N3 | The order shown correctly | It’s a snapshot of what’s already displayed in the order list, so I expect it to be identical | As expected |  |  |
| N4 | Confirm button clicked | The receipt image is generated and saved to the desktop with a message to the user giving the filename, the dialog closes and back to the intro screen | As Expected |  | Pass |
|  |  |  |  |  |  |
| Exceptional | | | | | |
| E1 | Unable to read the desktop folder from the registry to save the receipt image | Error message shown giving the customer their order number as a backup for the receipt printing error | As expected |  | Pass |
|  |  |  |  |  |  |

Source code and video demo

Right click -> Open Link to view the video.

[](https://www.youtube.com/embed/p765zu1yKlI?feature=oembed)

Recommendations

Allowing the customer to scan a QR code to install a mobile version of the application, further features could be added. For example, an order status system where the kitchen could update the status as the order is being prepared and cooked. It might be considered to make a mobile application solely for storing a digital receipt and showing the order status, keeping the order process tied to the kiosk only.