ACCESSIBILITY

Principles of Universal Design

1. **Principle 1: Equitable Use**
    a. Our current design does not have these features embedded but is compatible to system accessibility service features. We thought of implementing such features by hand like adding an accessibility service to our application but failed of unknown reason (We wrote these services but when deploy to the emulator, it show malfunctioning) so we decide to make our program comply with the accessibility services provided by Android. In Android system, you can use Setting->Accessibility and enable Talkback screen reader. This feature is usable with our appliacation as our application takes all the strings using Android:string, a type identifiable by the system. Under the framework of Android, it can use the talkback feature to describe the scree and all the texts for people with sight disabilities. Furthermore, as the literals are in Android:String, Android also provides the functionality to translate the texts to target language. Apart from that, we used the system default background so when it is set in settings, a dark mode or high contrast mode is compatible for user to choose. Some features we could implement in the future include text to speech for those with vision impairments or difficulty using touch screen devices. We could also use voice

recognition (similar to Apple's Siri) that can listen to the user's commands to input what the user wants to order (allowing placing an order or leave a review using voice recognition).

2. **Principle 2: Flexibility in Use**
    a. There is currently no choice in method of use, only option. Our application uses the fonts, the style of default Android, so when user change their preferences like fonts, font sizes, and background colors, button sizes in Settings of the system, the appearance of the app will also change accordingly. This gives, to some extent the flexibility in use while a embedded setting for user preferences will do better. This can be changed by providing options where the user can select where they want the buttons to show up (e.g. on the bottom, on the left, etc.). Potential future extensions could also have different language options.
    b. The program does not adapt to the user's use. Future additions to fix this could include saving the customer's information (if they wish) so that things like delivery address for placing order, or customer name for placing review will be automatically filled in.
3. **Principle 3: Simple and Intuitive Use**
    a. There are no complex steps required for the user to complete an action. They only need to enter the required information. The information required is requested by clear prompts.

b. Some tasks do not provide immediate feedback during task completion. For example, if a user enters a delivery address when they have selected a dine in order, the system does not immediately warn them that the input is invalid, and only displays an error message (with no details about what the error is) at the end. This could be changed to clearly display exactly what was wrong with the order so the user can respond accordingly instead of having to guess what they entered incorrectly.

4. **Principle 4: Perceptible Information**

   a. The color of the buttons is clear against the background, and the text is also clearly displayed. We choose purple (RGB: 85, 32, 181) and blue (RGB:33, 150, 243) as the background colors of the buttons, which, have a contrast ratio of 9.29 and 6.87, much above the design guideline by Web Content Accessibility Guidelines (WCAG) of requiring at least 4 for text on background. Therefore, the design of colors in our application well represents our design in comply with perceptible information. Instructions and prompts are given to the user, so the user knows what to enter. For example, the main page prompts customers to navigate to the customer page, or for staff to login.

   b. There is currently not much accommodation for people with sensory limitations. That can be changed by having options for

changing the color of the buttons or size of the text to accommodate color blind or dyslexic demographics.

5. **Principle 5: Tolerance for Error**
    a. The dishes in the order can be edited if the user accidentally selects the wrong dish. The edit order functionality shows warning when user tries to edit an order that does not have any dishes added to it.
    b. The buttons could be made larger (and made to recognize when the use clicks very close to them). Editing functionalities could also be added for the placing review. Warning messages could also be added when the user inputs illegal arguments, such as a delivery address when the order is dine-in.
6. **Principle 6: Low Physical Effort**
    a. The user does have to move their hands to press the buttons and enter text, which is located on different parts of the screen for different actions, however we have tried to keep most of the confirmation buttons on the bottom of the screen.
    b. Some repetitive actions such as scrolling may have to occur when the user is selecting which dishes they wish to enter, so we could potentially add a search bar to search for a specific dish so the user doesn't have to scroll through all of the dishes to find the one they want.

7. **Principle 7: Size and Space for Approach and Use**
    a. Buttons are clearly displayed on the screen. Textboxes are also clearly displayed with a prompt. The user does have to use the entire screen of their device, however that is necessary to display all information in a visible manner.
    b. Other components of this principle do not apply as this program exists in a virtual environment, and this principle is about the physical space available for the user.

Program Target Market Audience

If we were to sell or license our program to customers, we would market our program toward small restaurants (not retail chains) who want to digitize their process of placing orders, processing orders, and managing their inventory. Since our current functionality does not separate dishes into categories, (and assuming we do not add this later) we would target smaller restaurants that do not have an excessively long menu, as customers would find it difficult to navigate with no clear organization. So our target is small restaurants looking for some digitization of their ordering process.

Whether or not your program is less likely to be used by certain demographics.

Our program does not have a text to speech functionality, so those vision impaired would be less likely to use our program. In addition, our program would not be used by anyone who doesn't order food from restaurants, as they would

not need to see the menu or place an order (the functionalities of our program are not useful for them). People who live too far away from the restaurant would also be less likely to program as the restaurant would not deliver that far so placing an order serves no purpose unless they wish to spend a lot of time travelling to pick up their food. In addition, people who use other third-party ordering websites such as UberEATS are also less likely to use our program as they can see the menu and order on other sites.