**Introduction**

After following the general download and set up instructions provided by Readme file, user is now able to see the web page interface. The queue system consists of three components that supports customer service, counter staff, and customer relations officer. Moreover, the queue system designed to support both in-person walk in and online booking via mobile phones. Customer queue request was designed to manage customers’ queue orders based on business types and priorities. Business types include private banking and corporate banking, which are the two main business types across banking industry. Private banking business line prioritizes customers based on customer’s age that senior citizens were given preferential treatment for services, while corporate banking business line prioritizes customers based on customers’ asset and wealth level. Upon successful input the required information to generate a queue number, the customer’s queue number record will connect to the backend server and update to the counter staff to display it on the main TV screen. Meanwhile, customer relations officer closely monitor the entire queue list on its own monitor interface for the respective categories and stand by for any circumstances that requires stop/re-initiate the queue of either queue category.

**Detailed Steps Illustration**

**Customer:**

Graphical user interface, text, application, Teams

Description automatically generated

Figure 1: Demo of In-person Queue Generation Page

As shown in Figure 1, the interface of in-person walk-in queue generation front end development was designed by HTML and CSS. Information of business types and priorities were structured under an ordered list section, where each list provided selection choices. Selection labels were incorporated in this section that the users can select the answer applied to them. The main header – generate your queue number -- was designed to have 36 pixels font with navy blue colour (#005AA7) and placed in the upper margin of the page that customers can easily understand the function of this page. A getQbutton displays -- get the queue number – was incorporated and designed with 16 pixels font and aligned with the colour code of the main header (#005AA7).

To proceed, user is expected to select the labels provided for required inputs followed by pressing the Get Queue Number button. Upon successful completion of generating the queue number, a queue number report will forward to the customer as shown in Figure 2 (please regard that the branch is subject to change with your current walk-in branch):

Graphical user interface, text, application

Description automatically generated

Figure 2: Queue Report

The queue report was designed by html, the main header – queue report -- with 36 pixels font and placed in the middle margin of the web page. The main header’s colour aligned with previous webpage’s header colour (#005AA7). Under the main header, customers’ current queue number; business type; and branch were displayed in header 3 with universal 18 pixels in size and grey colour (#5F5F5F).

However, as shown in Figure 3, without fulfilling the requirement to select all required inputs, the system will prompt users to insert input again. A Go Back button provided for user to press and return to the queue generation web page.

Graphical user interface, text, application

Description automatically generated

Figure 3: Fail to Input Page

Fail to input page was designed with a main header displays the message of “Please input all the required information” to alert user that he/she did not follow the correct protocol to generate the queue report. The main header was designed with 36 pixels in size and placed in the middle margin of the webpage. A goBack button incorporated to allow user press and go back to the previous page.

Assume user successfully input all the necessary information and generate the queue report. The backend server will record and retrieve user’s input and store it in the backend server for further actions.

Customer queue number generation service also designed on the mobile phone platform as shown in Figure 4. Its front-end webpage is similar to the walk-in queue generation system except that the mobile platform provided a drop-down selection button for users to decide which branch he/she intend to go.

Graphical user interface, text, application, Teams

Description automatically generated

Figure 4: Mobile Queue Generation System

Mobile queue generation system follows same base design as the figure 1, except that the mobile platform incorporated branch selection that customers can select branch from Hougang, Jurong Point, Jurong East, and Kuala Lumpur. These sections record will be retrieved and stored by the backend server after user typing the Get Queue Number button.

After successfully generating the Queue number, a same queue report as the figure 2 will be displayed on user’s interface. See below figure 5:

Graphical user interface, text, application

Description automatically generated

Figure 5: Queue Report on Mobile

Similarly, when users failed to input all required fields, a same system prompt will ask user return to the previous page as shown in Figure 3. See below Figure 6:

Graphical user interface, text, application

Description automatically generated

Figure 6: Fail to Input Page on Mobile

**Counter:**

After queue generation process, the backend serve recorded all the user queue number records. Counter staff could utilize the backend server record to monitor the ongoing queue calling process. Counter staff given right to proceed with queue calling; skip the current queue number if the customer is no longer present in the branch; or at certain circumstances to pause the current queue calling process.

Graphical user interface, application, website

Description automatically generated

Figure 7: Counter Service Interface

As shown in Figure 7, counter service interface was designed by HTML and CSS. Main headers displayed the branch name, business type, and the counter for service. Main headers were designed with 30 pixels in size and placed on the upper margin of the webpage, headers colour aligned with the universal navy-blue header colour (#005AA7). The second header displayed the current serving number, which is the queue number that customer hold. The second header had 24 pixels in size with grey colour (#5F5F5F). Three buttons of Next, Skip, and Stop incorporated for counter staff to type, and they were designed in navy blue colour (#005AA7).

Once the counter staff decide to press any of these three buttons, the backend server will retrieve and respond accordingly. For instance, if counter staff pressed Next button, the serving number will then move on to the next in the queue; Skip button will skip the current serving number to the next customer waiting in the queue; Stop will pause the next queue number calling process until counter staff re-initiate with Resume Serving button.

Text

Description automatically generated with medium confidence

Figure 8: Counter Stop Serving Button

Counter stop serving button, as shown in Figure 8, the main header with message of Stop Serving set to have big font and placed in the middle of the webpage. This was designed to alert the counter staff that the current queue system has paused. A Resume Serving button was incorporated below the main header, counter staff could press the button to return to the previous counter page and resume calling as shown in Figure 7.

A picture containing timeline

Description automatically generated

Figure 9: Main Display on TV for Counter

Assume no pausing by the counter staff. Customers sitting in the main lobby waiting to be called will access to this TV display that illustrate the current serving customer queue number and its related service counter according under each business type, as shown in Figure 9. The output of the display designed to sit in a tabular form. Two tables were incorporated in the HTML design, break line inserted between these two vertical tables. All texts from both tables set to centre text align for better visualization. Now serving and counter were designed in the same rows that customers could easily check and interpret the serving number and its related service counter. A separated row indicated the missed queue numbers that customers could check if his/her queue number was skipped and talk to the customer relations officer for further actions.

Essential headers such as business types, now serving, counter, and missed queue numbers were padding with the universal navy-blue colour as previous pages (#005AA7) along with white font text. This design will allow the customers to pay attention to essential information listed on the screen. The counter main display will keep updated with the backend server to refresh and continue facilitate the queue calling procedure.

**CRO :**

Customer relations officer have his/her own user interface, and closely monitor the entire queue process. Customer relations officer also deal with missed queue customers by adding their missed queue number back to the queue list. Under certain circumstances, customer relations officer given rights to terminate the service.

A picture containing timeline

Description automatically generated

Figure 10: CRO Interface

The CRO display will show up the entire queue list, including current serving numbers, waiting numbers, and missed queue numbers. This interface was made by HTML & CSS. Information displayed in a tabular form, which is similar to the Figure 9 counter TV display. Two tables added under HTML body section, each of them separated by break line. Now Serving and Counter tables rows are placed in the same row, where customers can refer their queue number to the right counter. Waiting Queue table row was added to allow the CRO check how many customers waiting in queue. Waiting queue number according to the business type provided under the Waiting Queue section. A Missed Queue Numbers table row added to display missed queue numbers and CRO can decide to add those missed queue numbers back from add queue number interface. Business types, serving number, counter number, waiting queue number, and missed queue numbers of rows were highlighted by navy blue colour (#005AA7) with white text font. All texts under both tables aligned in the centre.

The main header System Status in 36 pixels font size with navy blue colour (#005AA7) placed under those two tables. Two buttons of Terminate Service/ Reinitiate Service added under the main header. CRO could press Terminate Service to stop the queue calling process or press the Reinitiate Service button to continue the queue process. Backend server will update the display accordingly.

Graphical user interface, text, application

Description automatically generated

Figure 11: System Terminated Interface

Once the CRO pressed Terminate Service button, a message indicate that system had terminated will pop up as shown in Figure 11.

Text

Description automatically generated

Figure 12: Termination Message

As shown in Figure 12, this interface will pop up after CRO pressed the button of Terminate Service. The main header with text “The Queue System has been terminated” in 36 pixels size with navy blue colour (#005AA7), placed in the centre margin of the page. A smaller header with text “Please try again later” placed under the main header. The smaller header designed in grey colour (#5F5F5F) with size in 20 pixels.

Graphical user interface

Description automatically generated

Figure 13: Reinitiate Service Interface

As shown in Figure 13, the system will back to available when CRO pressed Reinitiate Service button.

Graphical user interface, text, application, email

Description automatically generated

Figure 14: Add Missed Queue Number

One of the CRO responsibilities is to add missed queue customers back to the waiting queue list. Therefore, an add missed number into queue interface was designed as seen in Figure 14. An ordered list created under HTML body section that consist of business types and priorities choices. Labels were also created for CRO to click the relevant option that applied to it. A number insertion cell was created to allow CRO insert the missed number. An Add button was created and placed on the under-left margin of the webpage, information will post to the backend server once CRO pressed the button.