Restauration Desktop, Mobile App, & Website: A Brief Technical Guide

For maintaining and adding features into the existing application on both mobile and desktop side

Written by: Group 14: Jan Matthew Miranda, Eric Jiang, Christian Remolado

Table of Contents

Downloading and Installing

Downloading the Application *Prerequisites involve having Android Studio/Eclipse/MySQL Connector installed. You can download them here: https://developer.android.com/studio/install.html https://www.eclipse.org/downloads/ https://dev.mysql.com/downloads/connector/j/

Direct to Computer:

1. On Computer, go to shared Google Drive Folder (Team 14) enter subfolder (1_code): Go to subfolders: Mobile Application and Desktop Application

2. Download the zip file (restAuto-master.zip) as shown in image below

3. Unzip file and import project into Android Studio *Prerequisites involve having Node.js installed. Recommend to download from https://nodejs.org/en/download/

Downloading the Backend Server (mobile)

1. On Computer, goto shared Google Drive Folder (Team 14) enter subfolder (1_code) and subfolder (Mobile Application)

2. Download the zip file (raBackend-master.zip) as shown in image below

3. Unzip the file to the desired folder

For website:

Download Apache 8.5 and run as localhost

Go to Manager App and click on “select war file to deploy”

Select the war file (restaurant.war) from the Google Drive folder

Deploy the file and go to localhost:port/restaurant

Alternative: Import war file into eclipse JEE or other similar IDE and download apache 8.5 from http://tomcat.apache.org/download-80.cgi , then run the index.jsp file on the server.

Working with the Program

Starting the Backend Server (mobile):

1. Open the folder raBackend-master and open the file index.js using you favorite text editor (terminal editor if your into that)

2. On lines 6-9 in index.js, replace them with the following:

host: "restauto.c8kfv5fb1sng.us-east-2.rds.amazonaws.com", user: "restauto", password: "restauto1", database: "restauto" It should now look like the image bellow

3. Open a command prompt that has node.js, for this guide we use the Node.js command prompt

4. Enter the directory of where you extracted raBackend-master and enter it

5. Inside the directory, input into the command prompt: node index.js

6. The backend server is now live and ready to receive api calls from the mobile application

Setting the local mysql connector to proper classpath (desktop)

1. Open the desktop folder RestaurantAutomationMainSystem and open the file .classpath using you favorite text editor.

2. Locate the mysql-connector-java-<version number>.jar in the downloaded mysql-connector-java file.

3. On line 5 in .classpath, replace the path to local mysql-connector-java.jar

4. MySql is now connected after the new .classpath is saved

Working with the Mobile application:

1. Open Android Studio.
2. On the top menu bar, Click File, in the drop down menu click Open File or Project. Follow the correct directory of where you saved the restAuto folder.
3. Click the restAuto folder and click OK.
4. Create a new Android emulator by clicking Run ‘app’ on the right side of the top bar. The button is a green triangle.
5. Click Create New Device. Since we are simulating a waiter’s tablet, choose the Tablet category and the Nexus 9 tablet. Click Next.
6. Choose Nougat API level 25, for the sake of compatible code. Download API 25 if it is not installed on the machine.

7. Click Next, wait for the device to be downloaded. Once finished, click Finish.
8. To run the product, click on the Run ‘app’ button again. Choose the new Nexus 9 API 25 option. Click Run.
9. If the emulator asks to agree Android’s terms of service, click Agree & Continue.
10. Android Studio will take about 30-90 seconds to launch the initial restAuto activity depending on your device. Once started, the login page will appear.
11. For the username, enter: waiter1. For the password, enter: waiter1. Hit Login to continue.
12. The floor tab will appear next (all UI layouts can be referenced in the user documentation guide), and choose the appropriate floor and table to create a new order for. Then click Continue.
13. The menu tab will appear next, populated with items from the menu database table. Checkmark the appropriate menu items you would like to add to the order and the appropriate quantities for each item. Then click Order.
14. Repeat steps 12-13 for each new order.
15. The order tab will appear last, populated with the current order’s items, quantities, and prices on the first, top list. A progress bar connected with the database shows the progress of the current order if it is sent to the kitchen. The subtotal will appear for the current order on the bottom. All of the open order’s are on the second, bottom list. Click another order on the second list to update and view that order’s items and subtotal on the top list.
16. If the customer would like to pay an order at the cash register in the front desk, click the To Desk button. This will notify the POS system up in the front that an order has to paid in the front.
17. If the customer would like to pay an order immediately, click the Pay button. This will prompt you to either choose payment via Paypal or Credit/Debit card. Choose whatever the customer prefers. The Paypal option will open a Sandbox purchase option in the Chrome browser to complete the transaction. The Credit/Debit card option will open a new activity asking the customer to input their credit/debit card information to complete the transaction.

Working on the Desktop application:

1. Open Eclipse

2. On the top menu bar, Click File, in the drop down menu, click Import. In the next import menu open General folder and select Existing Projects into Workspace. Then hit next.

3. In the new window choose ‘Select archive file’ and browse for the RestaurantAutomationMainSystem.zip file then click finish.

4. Once Eclipse is done importing go to src -> controller -> FloorController.java and change line 39 url path to the local host’s user. Ex: user is admin -> /Users/admin/eclipse-workspace/RestaurantAutomationMainSystem/images/

5. The Desktop application is now ready to be run.

Working with the website:

1. Run the war file on an apache 8.5 server
2. Navigate through the header to the desired page.
3. The menu jsp page has an input next to each item that is initially set to 0. The user can change it and submit the form after they agree to pick up the food. This information is sent to the takeout.jsp page, which does some simple math and outputs the receipt for the takeout order. This page also creates a row in the takeout table in the database.
4. The Reservations jsp page is loaded with the reservations of that day. It also states how many reservations are allowed per hour. This is a variable that is defined at the top of the Reservations page, so it can easily be changed to fit the size of the restaurant.

The code is split into different chunks of the if else chain that depends on the form hidden input named submit. Therefore, if submit = “create” then the create block of code will run and create a reservation, just like a switch statement.