

# Hungry Leopards

Kai Zhao, Olivia Deng, Johnny Huang, and Mira Yun

Department of Computer Science and Networking  
Wentworth Institute of Technology  
Boston, MA 02115, USA  
{zhaok, dengj1, huangj8, yunm}@wit.edu

## *Abstract*

Online order application is one of the most popular methods to order foods. Not only for commercial restaurants, most of higher education institutes also provide campus-dining applications. However, existing school applications are passive information-based solutions by providing menus, hours of operation, and location of the cafeteria or links to ordering applications. In this paper, we propose the *Hungry Leopards*, an android platform mobile application that provides not only online ordering and payment, but also a secure pick-up solution. By providing two different solutions for client side and business side, *Hungry Leopards* provides a complete online ordering system for both customer and business.

**Keywords** –*Online Ordering App, School Café, QR code, Undergraduate Capstone.*

## I. INTRODUCTION

Online ordering is now a popular and convenient way to order foods. For example, Panera Bread’s online order application offers consumer for both pick-up and delivery service to reduce time to queue up for shopping and checkout. Moreover, online paying system provides a more efficient and secure payment method where it reduces the risk of the money being lost or stolen, compared to carrying cash which is relatively inconvenient that the return changes may be too heavy to carry around. Online ordering and payment systems have been common recently, however it is hard to find a secure pick-up system deployed. Panera Bread’s online order application, for example, does not provide a secure pick-up system, where the orders are at the pick-up area for customers to pick up freely that might increase the risk of mistaken other people’s order.

Not only for commercial restaurants, most of higher education institutions also provide campus-dining services and their mobile or web applications to assist customers’ demands. Brown Essential is an application made by Hector Morales that allows Brown University students to check laundry availability and cafeteria menus [1]. Yale Dining is an application made by Yale University. It provides Residential Dining Hall occupancy information where it estimated current seats in use based on transaction data [2]. It offers a view of menus for the current date and the following date, a map to all residential and retail location, dining hall contact information, view of nutrition information and ingredients of the meal, and a link for 2Go Orders for eat-in or takeout [2]. Freshman 15-Umich Dining is an application made by Totem LC for the University of Michigan dining hall. It provides information on the hours of the cafe, address, menu, nutrition, and price. Additional features are sorting dining hall by distance, count calories, carbs, and protein by tracking meals, and a link to the navigation application for a route to the dining hall [3].

Existing school applications are passive information-based solutions by providing menus, hours of operation, and location of the cafeteria or links to ordering applications. In this paper, we propose the *Hungry Leopards*, a mobile application that provides not only online ordering and payment, but also a secure pick-up solution. Unlike other universities, Wentworth Institute of Technology (WIT) only has one cafeteria. There will always be long lines waiting to get the food and after that, there will be another long time for paying the food. For some other people, they traveled from other buildings to the cafeteria just to check out the meal they are providing. If they do not like the meal provided, then it can be time wasted by traveling back and forth. The rest of this paper is organized as follows. Section II describes the overall system of *Hungry Leopards* and implementation details. Section III concludes our work.

## II. HUNGRY LEOPARDS

*Hungry Leopards* is an android platform mobile application for all the students, faculty, and staff of WIT. Clients can view the menu ahead of time, so that they can decide on whether they like to travel to the cafeteria or not. If they like the food that cafeteria had prepared, they can use our *Hungry Leopards* to place an order and pay. When picking up the order, all users will have to scan the QR code that was provided to user after placing the pick-up order for verification purpose. Even *Hungry Leopards* is designed for WIT, our design and implementation can be used for all other institutions.

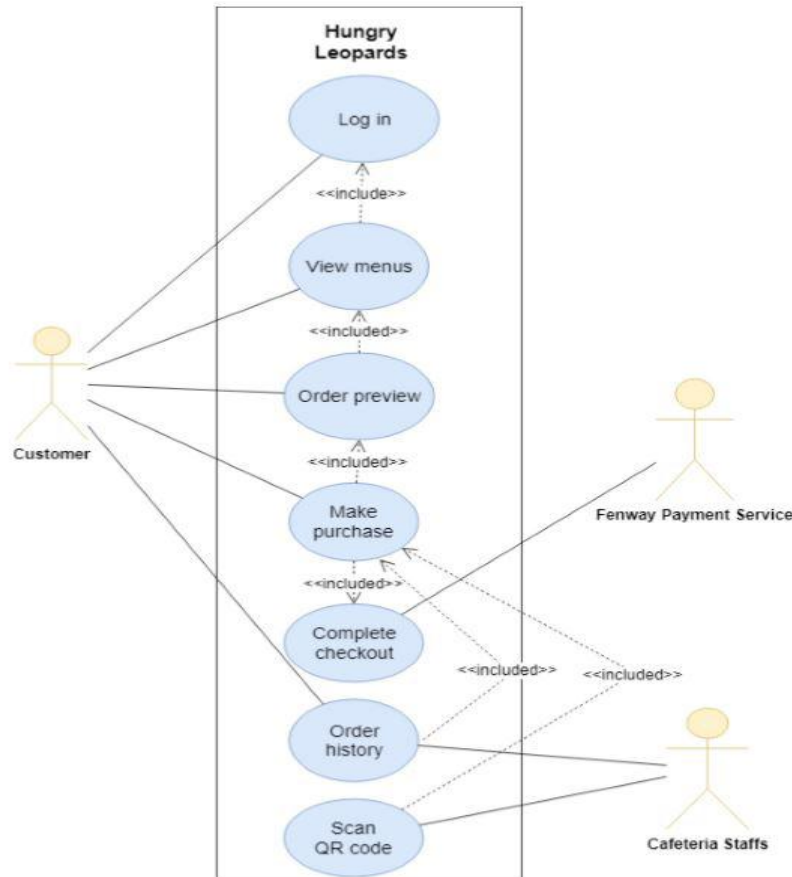


Fig.1. User Case Diagram

As shown in figure 1, *Hungry Leopards* has two sides, the client-side and the business side. For the client-side, the user needs to log in using WIT credential. After logging in the user can view the menu, order history, account balance, place an online order, make a payment, receive a notification when the order is ready, and being able to receive a QR code. For the business side, cafeteria staff can view the order that was placed on the client-side, being able to notify the client-side when the order is done and being able to scan the QR code for verification.

Figure 2 shows the ERD of our database. All students, staff, and faculty have their unique IDs for client-side to recognize users individually. So that everyone keeps their own order and account information within their account. When the order is placed on the client-side, the unique order number generated for the business side to recognize.

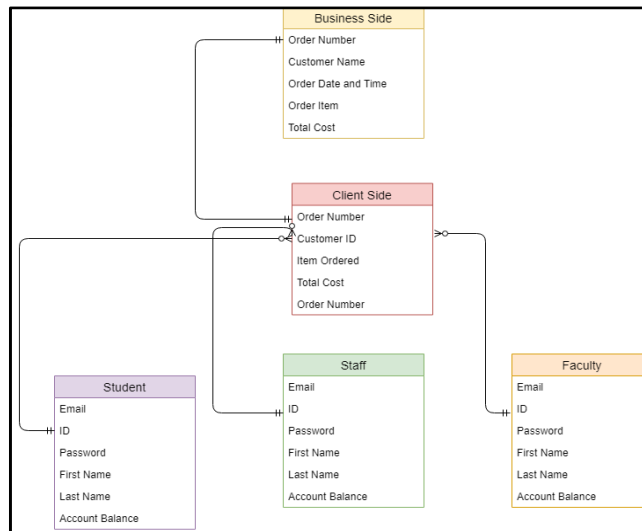


Fig. 2. Database ERD

## 2.1 Client Side

The client-side provides following functionalities: log in, view the menus, make a purchase, review the order, check account balance and pay, receive a QR code from business-side, and being able to be informed when order is ready.

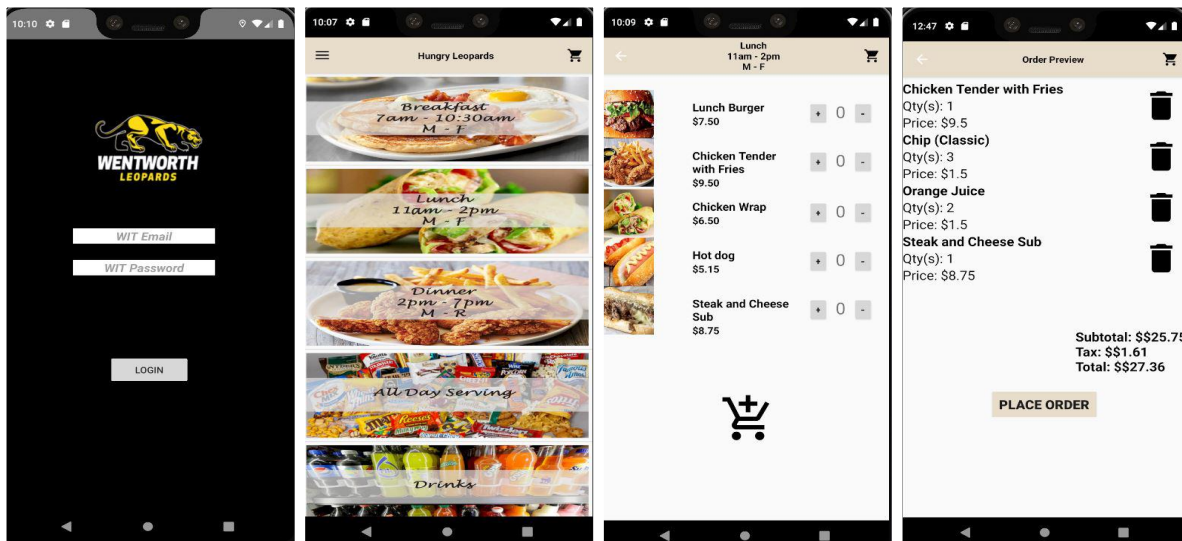


Fig. 3. Log in, Home, Lunch, and Review Page of *Hungry Leopards*

*Hungry Leopards* used the email sign-in authentication of Firebase [4]. As shown in Figure 3 Left, *Hungry Leopards* ask the user to input ID (WIT email address) and password. If it is successfully logged in, *Hungry Leopards* will bring the user to the home page otherwise it will pop out a message saying authentication failed. As shown in Figure 3 Second Left, Home Page is categorized to be breakfast, lunch, dinner, all day serving, and drinks. Each button can bring the user to their corresponding pages with the list of items using the position of the listview.

Figure 3 Third Left shows the lunch page with all serving time, menu, and price. The layout for breakfast and dinner are the same as lunch page and they all have a time restriction. User can only order food within a certain time range. As shown in Figure 4, if the user tries to order outside of the time range, then it will give a message saying “ the category is not available at this time”. When the add button is clicked, the selected item will be added to the current order page.

```

long cMillis = System.currentTimeMillis();
SimpleDateFormat sdf = new SimpleDateFormat( pattern: "HH:mm:ss");
Date cdate = new Date(cMillis);
String time1 = sdf.format(cdate);

String[] tokens = time1.split( regex: ":");
int secondsToMs = Integer.parseInt(tokens[2]) * 1000;
int minutesToMs = Integer.parseInt(tokens[1]) * 60000;
int hoursToMs = Integer.parseInt(tokens[0]) * 3600000;
final long total = secondsToMs + minutesToMs + hoursToMs;

String lMills = "11:00:00";
String[] tokens2 = lMills.split( regex: ":");
int secondsToMs2 = Integer.parseInt(tokens2[2]) * 1000;
int minutesToMs2 = Integer.parseInt(tokens2[1]) * 60000;
int hoursToMs2 = Integer.parseInt(tokens2[0]) * 3600000;
final long lowerTotal = secondsToMs2 + minutesToMs2 + hoursToMs2;

String uMills = "14:00:00";
String[] tokens3 = uMills.split( regex: ":");
int secondsToMs3 = Integer.parseInt(tokens3[2]) * 1000;
int minutesToMs3 = Integer.parseInt(tokens3[1]) * 60000;
int hoursToMs3 = Integer.parseInt(tokens3[0]) * 3600000;
final long upperTotal = secondsToMs3 + minutesToMs3 + hoursToMs3;

```

Fig. 4. Lunch Page Code

Figure 3 Right is an overview of all the selected item, with the total price calculated. As shown in Figure 5, when the pay button is clicked, it will compare the total price with the account balance. If the account balance is enough to make a purchase, it will create a new table to store all data information of this order with an order number randomly generated. It also will update the balance in the user's database. If the account balance is not enough, then it will pop out a window asking the user to check their balance.

```

db.collection( collectionPath: "TempItemStore").get()
    .addOnSuccessListener((OnSuccessListener) (queryDocumentSnapshots) → {
        for(QueryDocumentSnapshot documentSnapshot: queryDocumentSnapshots){
            String foodName = documentSnapshot.getId();
            int foodQty = documentSnapshot.getLong( field: "Qty").intValue();
            double foodPrice = documentSnapshot.getDouble( field: "Price");
            String Preparing = "Preparing your order";
            map.put( k: "Price", foodPrice);
            map.put( k: "Qty", foodQty);
            map2.put( k: "Date", currentTime);
            map2.put( k: "User", Login.emailID);
            map2.put( k: "Status", Preparing);
            map3.put( k: "Date", currentTime);
            map3.put( k: "User", Login.emailID);

            if(accountBalance >= ftotal) {
                db.collection( collectionPath: "OrdersNum").document(Integer.toSt
                db.collection( collectionPath: "OrdersNumBusiness").document(Integ

```

Fig. 5. Order Preview Page Code

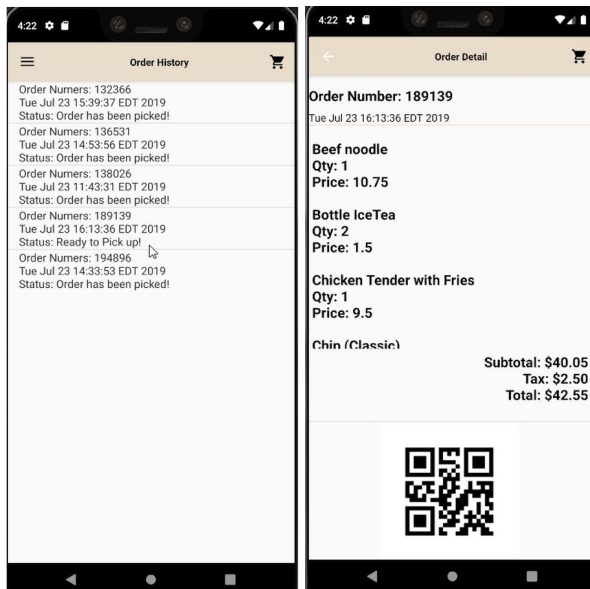


Fig. 6. Order History and Detail Pages

Order History Page gets its information from the database [5]. As shown in Figure 6 Left, it contains the order number, order date, and order status. When the order information clicked from the history page, it will open the Order Detail Page. As shown in Figure 6 Right, Order Detail contains all the information of the order. It gets data from databases by specific order Number which is passed from Order history class, and display them in the format of the item name, item quantity, and item price in a scroll view.

## 2.1 Business Side

The business side of Hungry Leopards provides three main functionalities for the cafeteria staff members: 1) receive the order from the client-side, 2) being able to inform the client when the order is ready, and 3) need to be able to scan the QR code for pick-up.

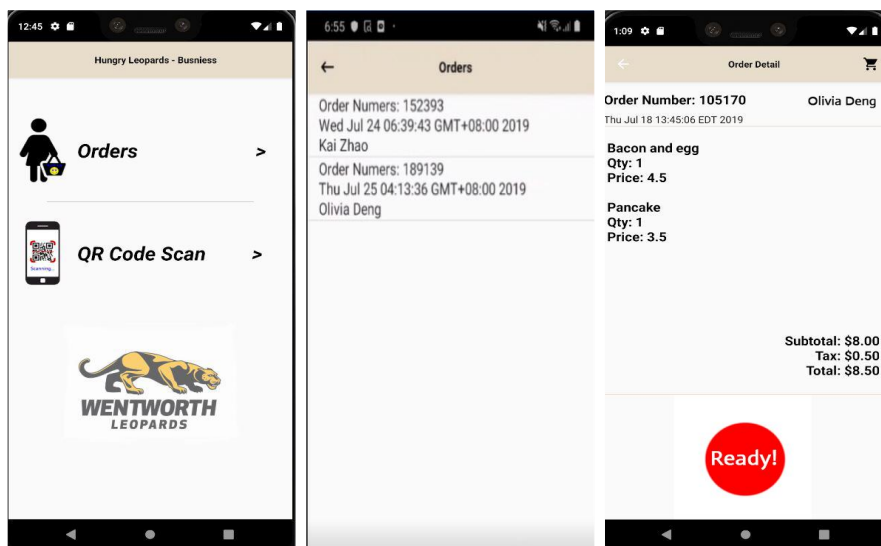


Fig. 7. Home, Orders, and Order ReadyPages

As shown in Figure 7 Left, business side provides two buttons, one goes to order details and the other one goes to scanning a QR code. Figure 7 Middle is the order list page. In this page, it shows all the orders that have been placed by the client-side. It passes order number and order date to order detail, by using global ArrayList that created at Homepage. When the order information clicked from the Orders Page, it will open the Order Detail Page. Figure 7 Right is the order detail page shows all the information of an order including the item name, quantity, price, and total price. If the ordered foods are ready, the the user of business side will click the Ready button as shown in Figure 7 Right. Then the client side application will receive this Ready notification.

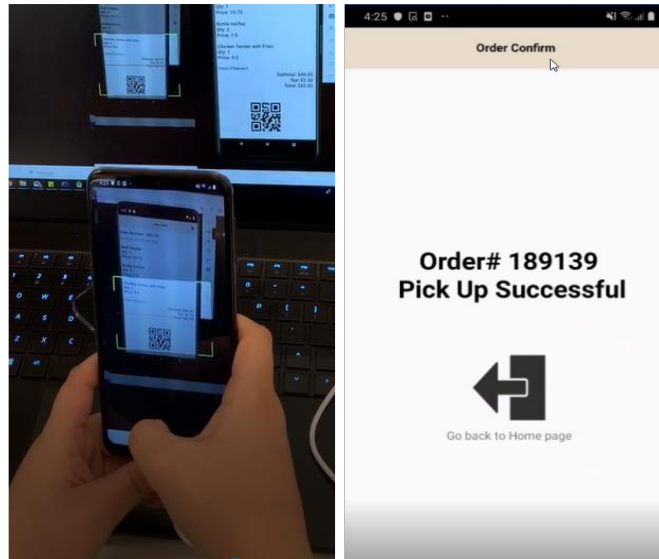


Fig. 8. QR Code Scanner and Pick-up Notification

If ordered foods are ready, clients can pick them up by using QR code generated from Order Detail page. The business side can provide a secure pick up method by using this QR code for opening pick-up box. In this prototype, we used camera flash light to verify the correct QR code. Even we have not implemented pick-up box or lock in this paper, our QR code method can be easily used and implemented with real locks or devices. Figure 8 Left shows our QR code scanner [5]. Figure 8 Right is the result after scanning the QR code if user's order had been successfully picked up.

### III. CONCLUSION AND FUTURE WORK

Lots of universities and schools provide their campus dining application. However most of them provides passive information on menu, location, and hours. Just like existing commercial restaurants, higher education institutions can also provide online ordering, payment, and secure pick-up solutions for all students, staffs, and faculty. In this paper, we have designed and implemented our campus online dining system, *Hungry Leopards*. *Hungry Leopards* is an android platform mobile application. By providing two different side, client side and business side, *Hungry Leopards* provides a complete online ordering system for both customer and business. In addition, we used QR code generator and scanner for secure pick-up methods. The future work of this research includes implementing pick-up box and extending the system to support easy adaptation from other institutes or organizations.

### REFERENCES

- [1] Morales, Hector. "Brown Essentials." App Store, 4 Feb. 2018
- [2] "Yale Dining - Apps on Google Play.", Google, Google, [https://play.google.com/store/apps/details?id=org.yaledining.app&hl=en\\_US](https://play.google.com/store/apps/details?id=org.yaledining.app&hl=en_US).
- [3] LC, Totem. "Freshman 15 - Umich Dining." App Store, 5 Jan. 2016 <https://apps.apple.com/us/app/freshman-15-umich-dining/id1069195670>.

- [4] Firebase Email Authentication, <https://firebase.google.com/docs/auth/web/email-link-auth>
- [5] W. Li, C. Yen, Y. Lin, S. Tung and S. Huang, "JustIoT Internet of Things based on the Firebase real-time database," 2018 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE), Hsinchu, 2018, pp. 43-47
- [6] Owen, Sean. "Zebra Crossing from the ZXing Project." Zebra Crossing, <https://zxing.appspot.com/>.