FoodPool
An Android Application to Pool Orders

Jay Chatwani¹, Ansh Shah², Stevina Correia³
¹,²,³ Information Technology, Dwarkadas J Sanghvi College Of Engineering

Abstract—Technology has become a vital part of human society. Various human activities have moved online, and there is a constant change in the way simple tasks are done. This has also affected the food industry. Food can simply be ordered on applications, without any hassle. Our application, FoodPool, aims to further improve this process by allowing people in the same area to pool their orders. This gives them a chance of meeting the high minimum amount required to avail discounts, thus saving users’ money, and at the same time, decreases the number of trips for the deliveryman. It is also a single-stop platform for users ordering food, where they can also browse through the potential discount offers, that are provided by various restaurants.

Keywords—food, orders, pooling, delivery, android application

I. INTRODUCTION
Ordering food online and via applications is one of the most popular use for the internet now, and this project aims to better that experience for users by allowing them to pool orders and thus save money on delivery charges. It aims to help them get access to coupons and use them, which they would not have been able to do previously. A lot of times, there are incredible offers on takeouts and online shopping sites. However, most of these offers require a high cart value, which one person alone cannot avail. These offers could have been availed had there been more parties on the receiving end. This is what we aim to achieve through our application. The primary motivation is to link people who are interested in ordering a similar item, while allowing all those people to get additional discounts on their orders, which they would not have normally. Thus, also allowing people to save money.

With this project, we aim at allowing users the liberty to avail most offers available, which they alone might not have been able to avail. The application also helps them suggest and decide an intermediary place where the two parties could meet, collect the goods and exchange the money. It can be used not only for ordering takeout, but can also be used on other e-commerce websites in a similar fashion.

II. APPLICATION FLOW

Login/Signup: Allows the user to either log in with an existing account or sign up by making a new account.

Input 1: User may enter their ID and password.
Output 1: If valid, they are granted access. If not, error message shown asking the user to enter correct credentials.

Input 2: User may create a new account by entering details such as name, email ID, password, etc.
Output 2: New account is created and user is redirected to login page to enter ID and password to login.
Selecting Restaurant: After identifying user’s location, the application displays the available restaurants around the user. User can then select a restaurant that they wish to order food from.

**Input 1:** User selects restaurant  
**Output 1:** After selecting the restaurant, the application moves to the ‘menu’ page.

Choosing the Restaurant: The user is shown the entire menu for the restaurant that they have selected. The user can then select the food items that they wish to order.

**Input 1:** The user adds items into the cart  
**Output 1:** The application moves to the ‘quantity’ page, where user can decide the quantity for the particular food item.  
**Input 2:** User can look at the available coupons for the particular restaurant.
Choosing the Restaurant: Choosing Quantity: Here, the user can choose the quantity of the food item that they wish to order. The final bill total will also be calculated by the application and displayed.

**Input 1:** User can choose the quantity for each food item that they are ordering.

**Output 1:** The application will move to the cart.

**Cart:** The user is shown the final cart with food items and cost.

**Input 1:** User inputs waiting time. This is the period of time that the user is willing to wait for a potential match.

**Output 1:** A timer with the user's waiting time starts counting down time.

**Output 2:** If a match is found, the user is notified of it and given an option to confirm order and the application moves to the 'bill' page.
**Bill:** The final total is displayed, after applying the discount that the matched pair is eligible for.

*Input 1:* The user clicks on ‘confirm order’

*Output 1:* The app moves to the ‘payment’ page.

![Payment Image]

**Payment:** The user is asked for their card details.

*Input 1:* The user’s card details.

*Output 1:* After the user is done entering card details, they can confirm it and the delivery location page is shown to the user.

![Payment Details Image]

**III. ALGORITHM DEVELOPED**

Since there was no existing application that did what our application does, we developed an algorithm. The main backbone of the application is matching two people in the same area. Thus, the algorithm that we developed focuses on four major things:

- **Region ID:** The user’s location is detected and classified into region IDs. Each region has a different ID. E.g. Andheri is 1, Vile Parle is 2

- **Restaurant ID:** Each restaurant in a region has its own unique ID. E.g. Dominos in Vile Parle has ID 0011, and McDonald’s in Vile Parle has ID 0014
Timestamp: The user enters their waiting time. The timestamp adds the waiting time to the current time and creates a bracket. For e.g., if at 11:15 am, the user enters a waiting time of 30 minutes, the timestamp will read: 11:15-11:45

Matching Status: Assigned to each user, the matching status will read 0 if the user is waiting for a match, and 1 if a match is found.

When a user enters their order with their waiting time, these four values are stored in the database. For a user to match with someone else, both must have the same region IDs, Restaurant IDs, and matching status. The timestamps need not be the same, but must coincide. This is done so that the people who match are from the same locality, order from the same restaurant, and can place their order within their decided waiting periods. The matching status ensures that the user does not match with more than 1 person, and that the user doesn’t wait for a match after already finding a match.

IV. FUTURE SCOPE

With this project, we aim at allowing users the liberty to avail most offers available, which they alone might not have been able to avail. The application also helps them suggest and decide an intermediary place where the two parties could meet, collect the goods and exchange the money. It can be used not only for ordering takeout, but can also be used on other e-commerce websites in a similar fashion. A lot of e-commerce websites offer discounts, but they are usually only available on a hefty bill, which is very difficult to get as a common man. Even when shopping, various shops have discount offers but require the customer to buy a lot to avail them, for example, Westside offers discounts on 4 shirts, which is impractical for a single customer. However, 2 people can pool their shirts to reach the required 4 shirts, and avail the discount, thus saving money, while simultaneously also not buying unnecessary number of articles. Thus, the concept of pooling to avail discounts and collectively saving money has unlimited applications, and all of them benefit the users and customers.

V. CONCLUSION

Cheap, fast internet access and easy to use applications have aided in the shift of a lot of common processes to the online platform. While ordering food online and via applications is common, this project proposes a technique to design and implement a model where people in the same area, ordering the same food, can pool their orders and thus, in the process, may be able to avail discounts and save money on delivery charges. A user performs all his transactions on the platform provided on the application, and can also look at potential offers, coupons, and discount options available for that particular restaurant on the application itself. The project also looks to help create a platform that is easy to use and convenient, thus making the process of placing orders simple and easy, while, at the same time, ensuring that the users save money as well.

REFERENCES