

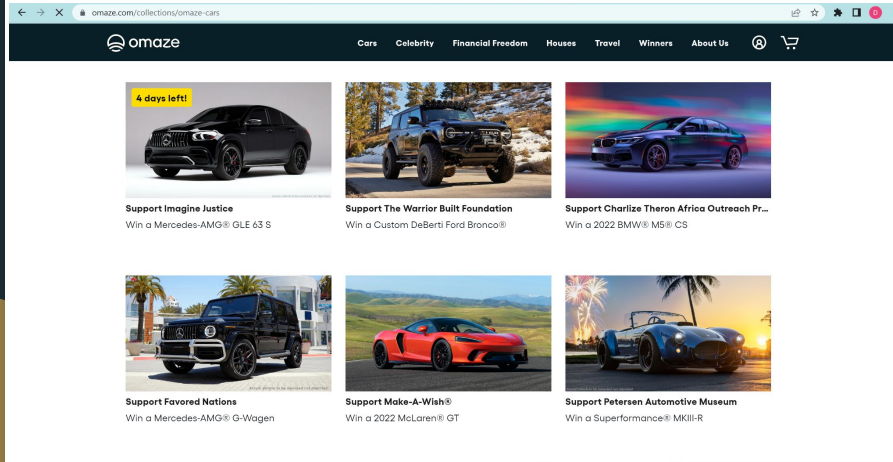


# Food Orders Made Easy with Code

By: Aryeh Stiber and David Kohanchi



# Our Original Idea: FREE CARS!



A screenshot of the Omaze website's 'Alternative Method of Entry' form. The form is titled 'Alternative Method of Entry' and includes a sub-header 'Give us your info:'. Below this, there are several input fields and checkboxes. The form also includes a 'Shipping address' section and a 'Submit entry' button. On the right side of the form, there is a summary of the entry, including the number of free entries, the prize, and the total cost.

**Alternative Method of Entry**

Give us your info:

All fields required, so we can let you know if you win (fingers crossed!)

First name Last name

Email address

☐ Want bonus entries, winner info, impact updates and more? Sign up for our email! It's not required and you can opt out at any time. For more information view our [privacy policy](#).

Free Entries \$0.00

Win a Mercedes-AMG® GLE 63 S

TOTAL: USD \$0.00

Promo code Apply

☐ I'm not a robot

Submit entry

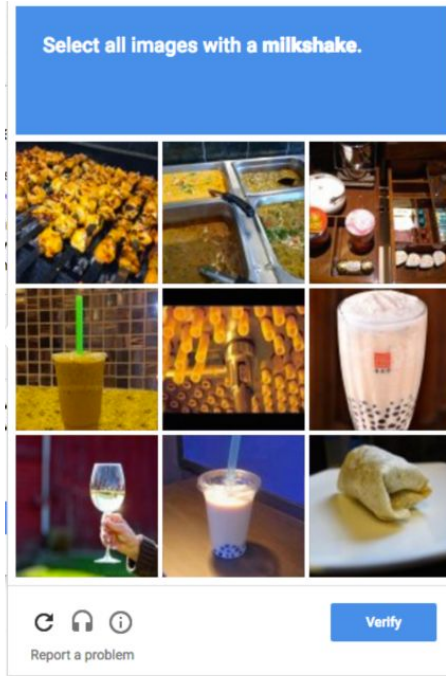
By continuing, you agree to our [Terms of Use](#), [Official Rules](#), [Privacy Policy](#) and [Refund Policy](#). For State Disclosure Information, [Click here](#).

Country/Region Åland Islands Zip/Postal Code

First input your information and watch the program enter you every 60 seconds (max number of entries).

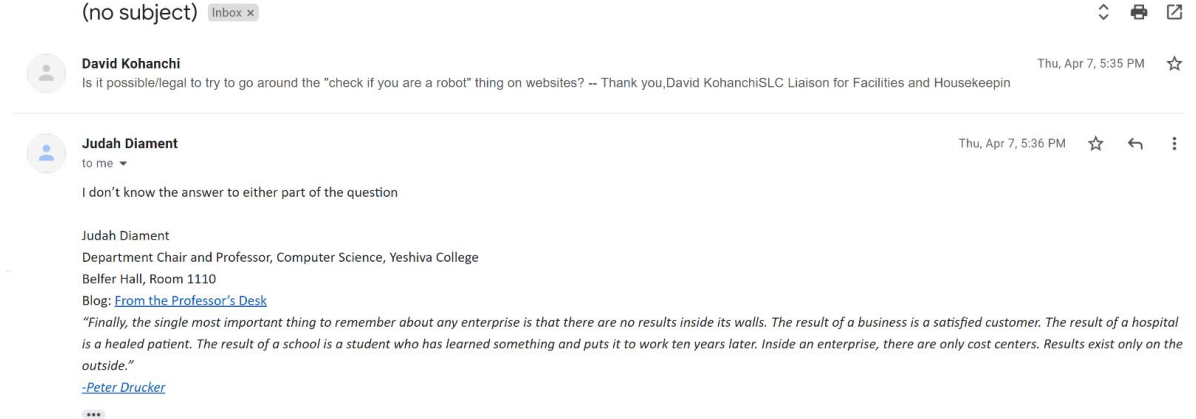
# Problems:

- Robot checkers!



# Solutions:

- Email Judah and quit



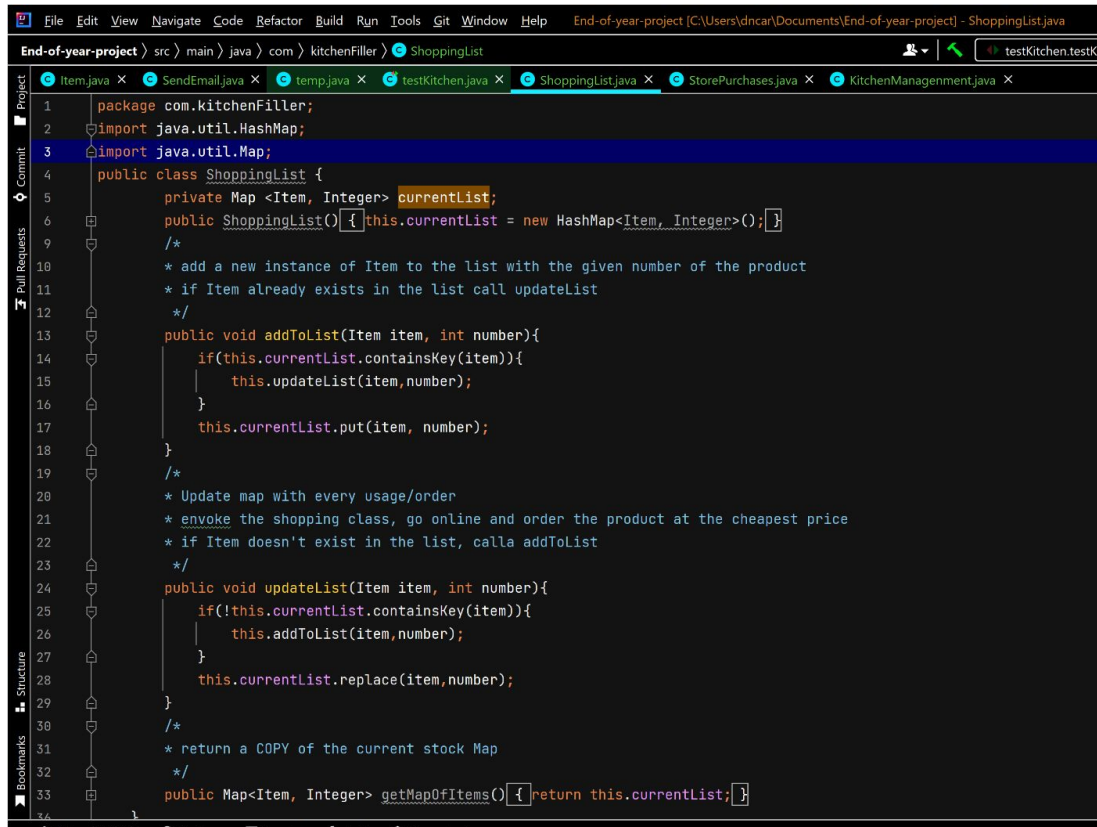
So on to our  
next idea!

# What we planned to do?

- Make a system in which a person can input an item that they want to purchase and how much of the item they want, and the item will automatically be ordered for them when they are low.
- Since this was particularly difficult and seemed weird that we would order an item without the person directly doing it, we decided that instead our system would send an email to the person ordering, which contained direct access to the websites that were offering what they were looking for and the page on the website in which they could purchase the item.

# Our Code

```
1 package com.kitchenFiller;
2
3 public class Item {
4
5     private String name;
6     private int weight;
7
8     public Item (String name,int weight){
9         this.name = name;
10        this.weight = weight;
11    }
12
13    /*
14    *Return the name of the item
15    */
16    public String getName () {return this.name;}
17
18 }
19
20
```



```
End-of-year-project [C:\Users\dncar\Documents\End-of-year-project] - ShoppingList.java
src > main > java > com > kitchenFiller > ShoppingList
Item.java x SendEmail.java x temp.java x testKitchen.java x ShoppingList.java x StorePurchases.java x KitchenManagenment.java x

1 package com.kitchenFiller;
2 import java.util.HashMap;
3 import java.util.Map;
4 public class ShoppingList {
5     private Map <Item, Integer> currentList;
6     public ShoppingList() {this.currentList = new HashMap<Item, Integer>();}
7
8     /*
9     * add a new instance of Item to the list with the given number of the product
10    * if Item already exists in the list call updateList
11    */
12
13    public void addToList(Item item, int number){
14        if(this.currentList.containsKey(item)){
15            this.updateList(item,number);
16        }
17        this.currentList.put(item, number);
18    }
19
20    /*
21    * Update map with every usage/order
22    * envoke the shopping class, go online and order the product at the cheapest price
23    * if Item doesn't exist in the list, calla addToList
24    */
25
26    public void updateList(Item item, int number){
27        if(!this.currentList.containsKey(item)){
28            this.addToList(item,number);
29        }
30        this.currentList.replace(item,number);
31    }
32
33    /*
34    * return a COPY of the current stock Map
35    */
36    public Map<Item, Integer> getMapOfItems() {return this.currentList;}
37
38 }
```

```
75 public String getLinkForArons(String product){
76     String link = "https://www.aronskissenafarms.com/search/";
77     product = product.replaceAll(regex: " ", replacement: "%20");
78     link = link + product;
79     return link;
80 }
81
82 private String getLinkForSeasons(String product){
83     String link = "https://www.seasonskosher.com/queens/#!?q=";
84     product = product.replaceAll(regex: " ", replacement: "%20");
85     link = link + product;
86     return link;
87 }
88
89 private String getLinkForWalmart(String product){
90     String link = "https://www.walmart.com/search?q=";
91     product = product.replaceAll(regex: " ", replacement: "+");
92     link = link + product;
93     return link;
94 }
95
96 private String getLinkForWasserman(String product){
97     String link = "https://www.wassermansupermarket.com/#!flushing/?q=";
98     product = product.replaceAll(regex: " ", replacement: "%20");
99     link = link + product;
100    return link;
101 }
```

Code that will be used to direct the user onto the website that they desire, specifically onto the page of that website where they can find the item they are looking to purchase.

## What we need to improve (given more time):

- Unfortunately there is something beyond our control that is incorrect in our code.
- We want to figure out a better way to send emails to the user from our code.
- Additionally, there are other features we'd like to add that would be able to find the price of the item on the website and list the choice of websites accordingly. Also, if we could find a way to report when a website is out of item, so the user won't receive it in their email.



# Sending Email Code

```
package com.kitchenfilter;

import java.util.*;
import javax.mail.*;
import javax.mail.internet.*;
import javax.naming.*;

public class SendEmail {

    private String email;

    public SendEmail(String email){
        this.email = email;
    }

    public void changeEmail(String newEmail){
        this.email = newEmail;
    }

    public String getEmail(){
        return this.email;
    }

    public void sendEmailFinal(boolean product) {
        // Recipient's email ID needs to be mentioned.
        String to = this.getEmail();

        // Sender's email ID needs to be mentioned
        String from = "web@psall.com";

        // Assuming you are sending email from localhost
        String host = "localhost";

        // Get system properties
        Properties properties = System.getProperties();

        // Setup mail server
        properties.setProperty("mail.smtp.host", host);

        // Get the default Session object.
        Session session = Session.getDefaultInstance(properties);

        try {
            // Create a default MimeMessage object.
            MimeMessage message = new MimeMessage(session);

            // Set from: header field of the header:
            message.setFrom(new InternetAddress(from));

            // Set To: header field of the header:
            message.addRecipient(Message.RecipientType.TO, new InternetAddress(to));

            // Set Subject: header field
            message.setSubject("You are viewing this on " + product.getName().toLowerCase());

            // Now set the actual message
            message.setText("You can order from the following websites: \n" + "1 - " + this.getLinkForOrder(product.getName()) + "\n" + "2 - " + this.getLinkForBasket(product.getName()) + "\n" + "3 - " + this.getLinkForWishlist(product.getName()) + "\n" + "4 - " + this.getLinkForAssess(product.getName()));

            // Send message
            Transport.send(message);
            //System.out.println("Sent message successfully....");

        } catch (MessagingException mex) {
            mex.printStackTrace();
        }
    }

    public String getLinkForOrder(String product){
        String link = "https://www.amazon.in/ref=cm_cr_brdp_w_cld_search/" +
            product + product.replaceAll(" ", "%20");
        link = link + product;
        return link;
    }

    private String getLinkForWishlist(String product){
        String link = "https://www.amazon.in/ref=cm_cr_brdp_w_cld_search/" +
            product + product.replaceAll(" ", "%20");
        link = link + product;
        return link;
    }

    private String getLinkForBasket(String product){
        String link = "https://www.amazon.in/ref=cm_cr_brdp_w_cld_search/" +
            product + product.replaceAll(" ", "%20");
        link = link + product;
        return link;
    }

    private String getLinkForAssess(String product){
        String link = "https://www.amazon.in/ref=cm_cr_brdp_w_cld_search/" +
            product + product.replaceAll(" ", "%20");
        link = link + product;
        return link;
    }
}
```