AN INTELLIGENT MOBILE APPLICATION TO DECREASE FOOD WASTE ACROSS SCHOOL CAMPUSES

Elaine (Yi-Hsun) Hsieh[1], Hailey (Yung-Chen) Hsieh[2] and Jenny Wong[3]

[1,2]Pacific American School, No. 307, Section 1, Xinglong Rd, Zhubei City, Hsinchu County, 30272
[3]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

In this research project, we aim to solve the issue of food waste particularly in school campuses after seeing the large amount of waste thrown away each day at the school lunch. Oftentimes, students dislike the food choices they are given, and some other times, food gets thrown away when they order more than needed. To combat food waste, we created FoodShare Connect which includes three different roles, school manager, family, and students. The school manager can add the food items for students to vote, as well as the food available for giveaways. This will effectively decrease food waste as the students get to vote for the food that they like, and the unused food products can be distributed to families who want it. Throughout the process, we resolved the issue of preventing users from signing up with the wrong role, dealt with data storage, and prevented students from voting more than once a day. This is done mostly with the use of Google Firebase, where we store the data, and Flutter language. We made sure to save all the necessary information on Firebase, check whether the user is using the same role to log in as their role saved in Firebase, and override all the voting information as the new day begins. Overall, we believe this APP can tremendously decrease food waste in school campuses, save a lot more for others in need, and most importantly, save our environment.

KEYWORDS

Flutter, Food Waste, Environment Protection, Mobile Application

1. INTRODUCTION

In this project, we're attempting to address the problem of food waste specifically in school campuses. According to the World Food Program, about 783 million people (10% of the world's population) go to bed hungry each night [2]. Similarly, the Food and Agriculture Organization states that approximately one third of food produced globally is wasted, which is about 1.3 billion tons of food [1]. This large amount of food that we toss away is enough to feed twice the number of undernourished people in the world [15]. Therefore, it is extremely crucial to take action against squandering the food resources we have. In supermarkets, workers would often stock a surplus of food then discard them once the food items aren't as fresh. In households, people would often buy larger packages that they aren't able to eat [6]. In the long run, preventing food waste can not only save money for individuals but also benefit the society as a whole. The food that would have been wasted could go to people that can't afford to purchase food. We've often seen the prevalent issue of food wasting in the community around us,particularly in schools

David C. Wyld et al. (Eds): SIPR, NCWC, BIBC, SOFEA, DSML, ARIA, NLP, CSEN -2023
pp. 51-61, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.131904

where leftover lunches are being thrown away. After seeing this, we were immediately motivated, and started to brainstorm various solutions targeting the food waste issue.

The US Department of Agriculture attempts to resolve the ongoing food waste issues in schools through letting students choose the lunch menu, extending lunch periods, and creating share tables [14]. Our project improves on this solution by creating an app that allows students to fill out surveys about the food that they want to eat and also allows families to view where there would be food items that they could get, allowing families to also be able to get food. In Alexandra Lagorio, Roberto Pinto, RuggeroGolini's research, it takes a stand against food overconsumption by decreasing the portions served to the students, providing nutritious food options, and making the food look more presentable. However, the solution didn't propose an effective way to decide how much food they should prepare. Our project improves the solution by creating a survey that would count how much the school should prepare. Others have also tried to compost extra food, however, this solution doesn't tackle the problem at its roots and we wish to do so through our app.

To tackle food waste, we decided to create a Campus Food Management and Conservation Mobile Platform, No Food Waste, to survey students' favorite food, decide the menu of future cafeteria food, and tally any unused or uneaten food. Our app would include three roles, the manager, students, and family. The students would be able to vote on the food items that they most want and fill in surveys. The manager would be able to see the food choices and number of votes by the students and decide the school menu. Family members would be able to see food items that are available and get the items. This solution will be effective because schools would be able to serve food that students like and at an adequate amount without having a lot of leftovers. Furthermore, this solution will also help low income families to be able to have enough food, decreasing food waste and benefiting many families at the same time. Other solutions that we've discussed are creating a sharing table for students to share unopened food items, reducing the amount of food provided by the school, or composting the extra food. Our solution would be better because we also added families so that families can also be able to get food, not just students, thus we can also help feed hungry families [7].

In the first experiment, we created 10 student accounts and used them to vote and 1 manager account to check if the voting statistics are accurate. In our experiment, we recorded the statistics for 10 days, checking the data stored in firebase and the voting statistics shown on the manager screen to see if the votes were accurately displayed. We found that our voting system is accurate, since all the votes were correctly displayed on all 10 days. In the second experiment, we created 5 family accounts and 1 manager account to test whether the family members can see the updated number of items if they logged in while the manager was adding food. We used the manager account to add food while we logged in the family accounts one by one, recording down the time they logged in and the number of food items they could see. The results revealed that family members could only see the food items that were added before they logged in, so the earlier they logged in, the less number of food they could see.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Bypassing the Wrong Login Role

In this application, we have to prevent different types of users from bypassing the wrong login role whenever users try to log in after creating an account. For example, if a student creates an

account, they should not be able to sign in with the manager or the family role. To prevent this, we plan to code a stopper that sends an error message whenever users try to sign in with the wrong role. Whenever a user is trying to sign in, the code will probably pull up information from Firebase and compare whether the role the user clicked on matches with the role they signed up for.

2.2. Saving All the User Information in a Database

We realized the need of saving all the user information in a database in order to display the necessary information. For instance, information about the manager needs to be displayed on the student and family page. Similarly, the student and family's information needs to be shown up on the manager's page, specifically the voting statistics. To resolve the issue, whenever the manager adds food items, we plan to save the information within a specific document name in Firebase. Our app will have access to Firebase, therefore, when students or family are viewing their screen, the code will access firebase and extract the necessary information through food or vote documents. Another problematic issue is also with displaying the voting and food data for the day only. Thus, we plan to resolve this by letting the code get rid of the information in the food document by simply changing the whole food document instead of appending information from day to day.

2.3. Accurate

In the app, students will get to vote for their favorite food each day in order for the school to prepare more of their favorite food, reducing food waste within campuses. However, in order for the voting statistics to stay accurate, we plan to implement a prevention system that works by preventing students from voting multiple times each day. Therefore, whenever the student votes on their screen, a few fields will be added to Firebase that display their vote as well as the day they voted. As such, whenever a student is trying to vote, the code will check if there is a vote key with the date set to the current day. If there is, then the system will prevent that student from submitting another vote.

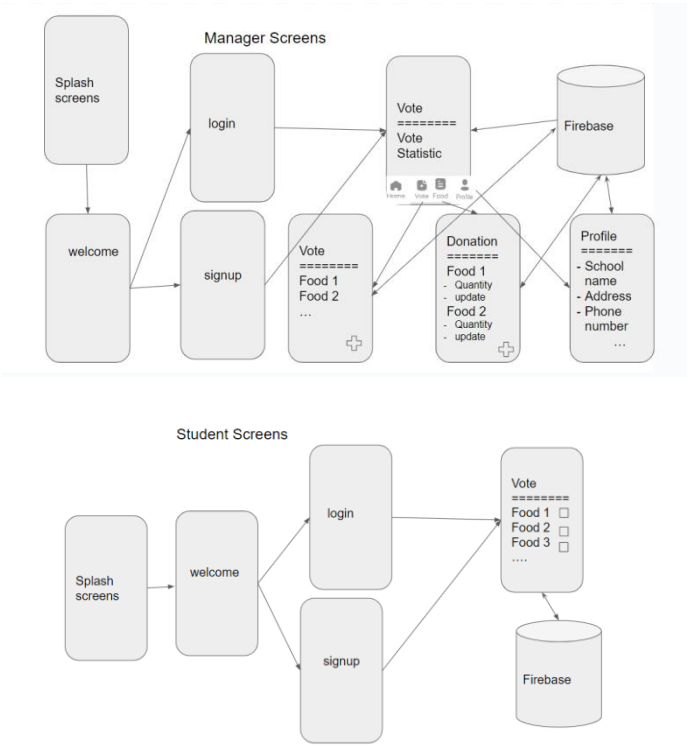
3. SOLUTION

In our app, there are three distinctive roles users can use to sign up with: the student, family, and manager role respectively. These three types of users each have access into their own screen after signing up or logging in. The manager will be able to see the vote statistics, for seeing the pie chart displaying students' votes; vote, for adding new food items for students to vote; donation, for listing food items good to be donated to family members; and the profile page, for logging out and displaying the manager's information. The family screen has two different functions, the home screen displaying the food items the school has to offer, and the profile page. Similarly, the student's page also has similar features, which includes the vote page, letting the students vote for their favorite food that the manager added for them.

Three main components in this program are the authentication service, database, and the vote system. The authentication service system is simply the backend service for the sign up and log in page. It is specifically for storing the user information into a database, so their account information can be stored and accessed when needed. The database, which is also our Firebase, is crucial in collecting all the information that the manager updates. Due to that, the family and student are able to access and view the information.Lastly, the voting system, where both the manager and student need to see the changes being made. The manager creates the survey, and is

able to see statistics when students voted, and students need to be able to see the choices the manager added.

The user information along with information about the food items and the number of votes for each item is stored in Firebase. We chose to use Firebase as it is free and has many services including Firebase Authentication service used to authenticate users. Furthermore, Firebase works under Google, so it has an extensive database for app development [14]. Firebase provides a realtime database that can even be used without internet connection. Lastly, Firebase provides a safe environment that allows us to set data permissions [8]. Our code is written in Flutter, which we chose due to Flutter's many customizable widgets that can allow designers to create beautiful app designs that are visually appealing. Furthermore, Flutter allows fast development since it allows developers to see their changes immediately after modifications. Moreover, Flutter allows smooth transitions between animations, even in older devices and is free for all users. Our app will be published on the App Store. When the app is opened, a splash screen will be displayed, then a short description about the role of students, managers, and family members. The user can then choose their roles and log in through the authentication service offered by Google's Firebase. When they log in, they are led to the separate home pages for different roles. There, students can vote for the food item they want, managers can see the number of votes, and family members can view the food items available.



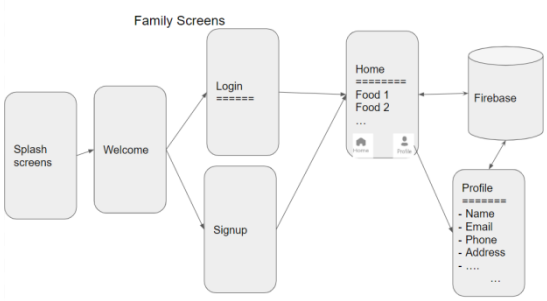


Figure 1. Overview of the solution

The authentication system is used to verify the users when they log in and ensure that they are logged in in correct roles. This component relies on the authentication system in Firebase to verify the users. Firebase authentication systems support various methods of logging in, including phone numbers, passwords, and even Facebook, Twitter, Google, etc. Our app uses the email and password of the user to log in [8].

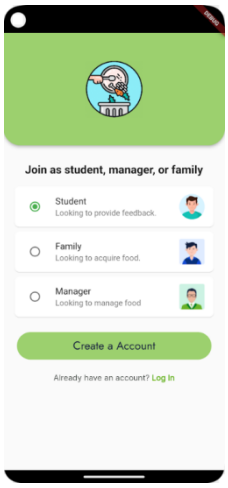


Figure 2. Screenshot of the APP 1

```
//SIGN IN METHOD
Future signIn(
    {required String email,
    required String password,
    required String role}) async {
  try {
    await _auth.signInWithEmailAndPassword(
      email: email,
      password: password,
    );
    Map<String, dynamic> info = await getUserInfo();
    if (role != info['role']) {
      return 'You must log in with the ${info['role']} role';
    }
    return 'successful';
  }
  on FirebaseAuthException catch (e){
    return e.message;
  }
}
```

Figure 3. Screenshot of code 1

This code segment aims to send out an error message whenever the user tries to use an invalid role to sign in, that is, when the role they used is different from who they are, otherwise, it will return successful. This code function is then implemented in the log in file, where the users are redirected to different user home screens according to their roles. The variables used in this section are the email, password, and role, which are all string variables used to store the parameters passed into the function in order to identify the user trying to sign in. Then, it compares with the user information obtained from the function getUserInfo. If the role the user is trying to log in matches the role they signed up with they would be able to successfully log in. Otherwise, the words "you must log in with the [user's role] role" will be displayed.

The database collects all the information that the manager updates, such as the food items for students to vote on and the available food for families. The database is highly important since all the information that needs to be displayed on the student and family screens all have to go through the database.

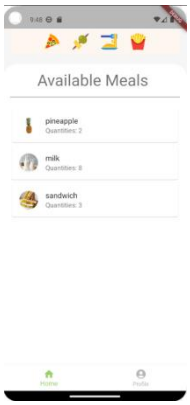


Figure 4. Screenshot of APP 2

```
Future<Map> getVote() async {
  Map<String, dynamic> data = {};
  await FirebaseFirestore.instance.collection(collectionName).doc('vote').get().then((DocumentSnapshot documentSnapshot) {
    if (documentSnapshot.exists){
      data = documentSnapshot.data() as Map<String, dynamic>;
    }else{
      print("document doesn't exist");
    }
    print("data:$data");
  });
  return data;
}
```

```
getData() async {
  final DateTime now = DateTime.now();
  final DateFormat formatter = DateFormat('yyyy-MM-dd');
  final String formatted = formatter.format(now);
  Map voteLog = await getVote();
  setState(() {
    if (voteLog.containsKey(formatted)) {
      foodOptions = voteLog[formatted]; // gets map of food items with info
      _food = foodOptions.keys.elementAt(0); // gets the first food item
      print(_food);
    }
  });
}
```

Figure 5. Screenshot of code 2

The getVote function creates an empty map named data and collects all the voting data from the vote document in the Firebase [11]. The if else statement checks if the document is empty or not, if the document isn't empty, the votes are stored into the data map, otherwise, the line "document doesn't exist" will be printed out. Then the data map is returned to be used in other parts of the code. In the getData function of the student screen, the first two lines get the date for today and formats it by year, month and day. Then, the getVote function is called to get the vote data and

stored in a map variable named voteLog. After that setState is called to update the two variables: foodOptions, which will get the entire map of food items for today and _food, which will be updated to the first food item from the database if the voteLog variable contains today's date, which is what will update the student screen.

The voting system aims to allow the students to vote on their favorite food item once a day to prevent repetitions and to allow the manager to be able to see the voting statistics displayed on their end and the number of votes for individual items [12]. This voting system again uses Firebase, which allows us to store the number of votes, the item, and the date.

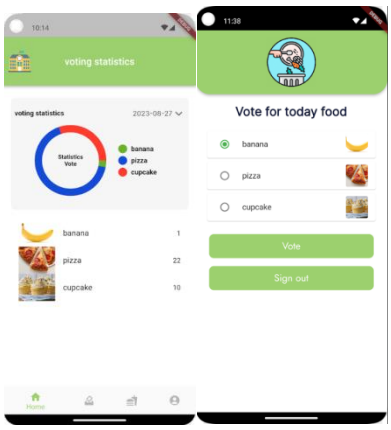


Figure 6. Screenshot of APP 3

```
vote() async {
  Map<String, dynamic> userInfo = await getUserInfo(); // gets user's info from Firebase
  foodOptions[_food!]['numberVote'] += 1; // adds a vote to the food item

  final DateTime now = DateTime.now();
  final DateFormat formatter = DateFormat('yyyy-MM-dd');
  final String formatted = formatter.format(now); // formatted date for today

  if (!userInfo.containsKey('vote') || userInfo['vote'] != formatted) {
    Map<String, dynamic> data = {formatted: foodOptions};
    userVote(data);
    userInfo['vote'] = formatted;
    editUserInfo(userInfo).then(
            (value) => ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
          content: Text(
            'Your vote was added',
            style: TextStyle(fontSize: 16),
          ),
        )));
  } else {
    ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
      content: Text(
        'You already vote',
        style: const TextStyle(fontSize: 16),
      ),
    ));
  }
}
```

Figure 7. Screenshot of code 3

The vote function gets each user's information and stores it in a map called userInfo. Then, the number of votes for that food item is incremented by 1. The if else statement checks if the user has already voted by checking if the word vote is in the userInfo map or if the vote document is in the proper format. If either of the conditions isn't met, the student would be able to vote since they haven't voted on that day yet. Then, the students' vote would be added and the message "your vote was added" would be displayed. Otherwise, if the student has already voted, the code would run the else statement. This would display the message "you already vote" and wouldn't allow the student to vote again. The information of these votes are all saved under specific time

of when the student voted in order to compare the time that the student is trying to vote with the data stored.

4. EXPERIMENT

4.1. Experiment 1

In this experiment, we create different student accounts and check if the voting statistics are correct after each student vote. This is important since the voting system is a large part of our app. The votes need to correctly be displayed so that the students will have their top choice of food.

In this experiment, we started a whole new emulator and created a manager account to add the different food items for students to vote. There are a total of six food items, including milk, pizza, granola bars, orange juice, spaghetti, and sandwich. For the experiment, we will then create 10 different student accounts and try to vote for different food items, documenting each vote as well to double check with data stored in Firebase, and the statistics displayed on the manager's vote statistics screen. All the votes need to be consistent throughout the various platforms to make sure the program is functioning correctly.

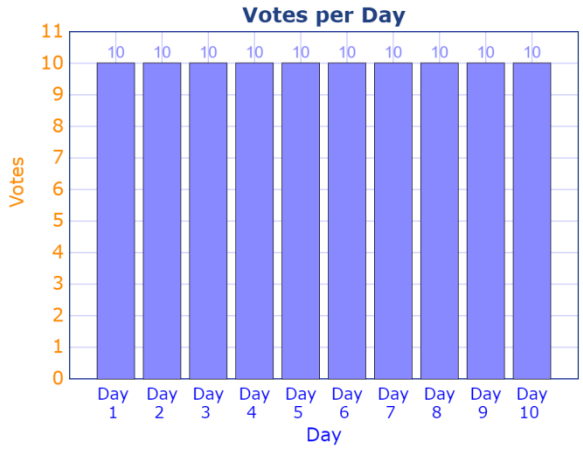


Figure 8. Figure of experiment 1

The results of the experiment are consistent with our hypothesis, that the number of votes from students will be the same as the number of votes that is displayed on the manager's screen. In the experiment, we let ten students vote for their favorite food item each day after the manager added in the food options. From the graph, it is shown that each of the student's votes got displayed on the manager's screen. After drawing the bar chart, we can see the accuracy of the code, which is likely a result of the working Firebase documenting every vote, both the student's screen and manager's screen linking to the database, and how our code prevented each student from voting more than once a day. Thus, none of the data surprised us. If we were to have a different number of students who voted each day, the resulting values on the y-axis will display that value instead.

4.2. Experiment 2

Another potential blind spot in our app is that if the manager adds food while multiple family accounts are already logged in, the family might not be able to see all the food items on their end without having to log in again.

In this experiment, we will create multiple family accounts and one manager account using multiple emulators. Then, we'll first use the manager account to log in and start adding food items. Once we've started adding food with the manager account, we'll log in the different family accounts at different minutes after the manager starts adding food and record down the number of items that we are able to see on each family account. We will then record the number of minutes after the manager started adding food that each family account was logged in and the corresponding number of food items that they can see.

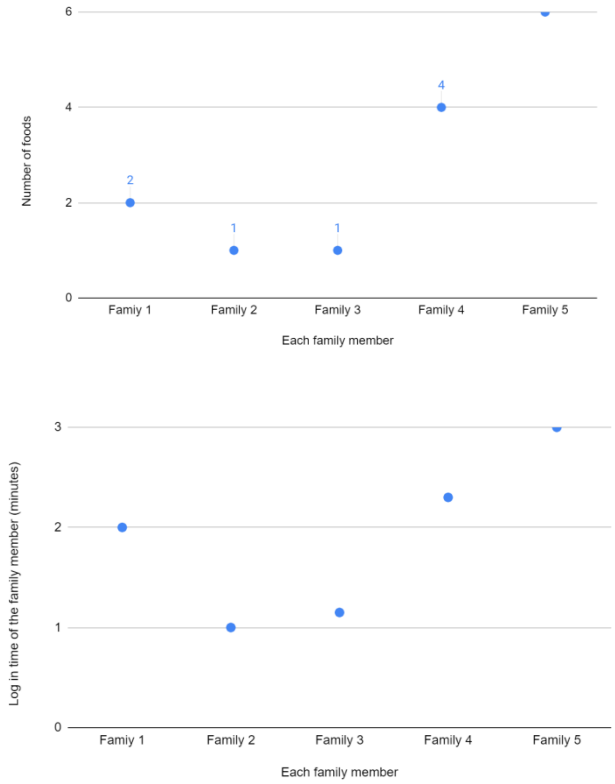


Figure 9. Figure of experiment 2

The results of the experiment are consistent with our hypothesis, that the number of food items that the family members are able to see correlates with the time that they log in (the number of minutes after the manager starts to post food items). In our experiment, family 2 logged in one minute after the manager started to add food and family 3 logged in just above one minute. At that time, the manager has only added one food item, so family 2 and family 3 can both only see 1 item. Family 5 logged in 3 minutes after the manager started adding food. By that time, the manager has already added 6 food items, so family 5 can see 6. The manager plays the biggest role in what the family sees, since the number of food items each family member can see is dependent on the number of items the manager added.

5. RELATED WORK

The US Department of Agriculture attempts to resolve the ongoing food waste issues in schools through letting students choose the lunch menu, extending lunch periods, and creating share tables [3]. Students could choose their lunch menu by holding tasting contests or creating a student committee to provide feedback about the acceptability of each food item. Schools are suggested to extend lunch periods since many students often don't have enough time to finish

their lunch. Lastly, the solution suggests schools create share tables for students to put food items that they don't want to eat to share with other students who may want an additional food source [9]. This solution seems highly effective since if students choose the food that they want to eat, it would likely decrease the amount of food wasted. Furthermore, share tables would allow unopened food to be eaten by other students. Our project improves on this solution by creating an app that allows students to fill out surveys about the food that they want to eat and also allows families to view where there would be food items that they could get. Our solution would also allow families that can't afford food to be able to have enough food.

In Alexandra Lagorio, Roberto Pinto, RuggeroGolini's research, it takes a stand against food overconsumption by decreasing the portions served to the students, providing nutritious food options, and making the food look more presentable [4]. However, the solution didn't propose an effective way to decide how much food they should prepare. Furthermore, by making the food look more presentable, students would be more inclined to eat more compared to food that doesn't seem as delicious. Our project improves the solution of decreasing the amount of food served by creating a survey that would count how much food is wasted and how much the school should prepare.

We noticed that others have tried to target food waste issues by creating composting sites, creating an inventory of the organic composts each day. According to researchers, 63 tonnesof food scraps can potentially be composted each day within UDC (campus of population around 20,000 people), with improper materials being well below one percent of the raw materials [5]. Therefore, composting can create an eco-friendly solution to the large amount of organic waste created by city life, and can create composts of high nutritional values. Although practical we believe targeting issues at its roots is a more effective solution. We recognize the need to reduce waste at its origins, also noting the fact that there might not be enough land to create compost sites in some countries' metropolitan areas.

6. CONCLUSIONS

One limitation to our project is that the manager has to manually add each item of food. This would be time consuming for the manager if they have to add many items. We could improve this by allowing the manager to choose from a list of foods to add or allowing them to add a spreadsheet with all the food they want to add. Further improvements is to have suggested images pop up when the manager adds information about each food item. This way they won't have to upload their own pictures each time they want to add an item, and it would also make the app look more professional and have a similar style. We could do this by creating a database with images of many food items. Moreover, we can improve the family screen by allowing families to put a hold on the amount of food that they want or allow family members to decrease the quantity of food that they took so the quantities would be updated.

Despite potential limitations regarding the usability of this application, we believe that it will greatly benefit the human population and help battle hunger as well as food wastefulness on this planet. In the future, as long as we fix the current issues of the app, it can be used more extensively across different schools and even in different countries.

REFERENCES

[1] De Angelis, Vanda, et al. "Multiperiod integrated routing and scheduling of World Food Programme cargo planes in Angola." Computers & operations research 34.6 (2007): 1601-1615.

[2] Torres-León, Cristian, et al. "Food waste and byproducts: An opportunity to minimize malnutrition and hunger in developing countries." Frontiers in sustainable food systems 2 (2018): 52.

[3] Purkey, Stewart C., and Marshall S. Smith. "Effective schools: A review." The elementary school journal 83.4 (1983): 427-452.
[4] Lagorio, Alexandra, Roberto Pinto, and RuggeroGolini. "Food waste reduction in school canteens: Evidence from an Italian case." Journal of cleaner production 199 (2018): 77-84.
[5] Torrijos, Verónica, Domingo CalvoDopico, and Manuel Soto. "Integration of food waste composting and vegetable gardens in a university campus." Journal of Cleaner Production 315 (2021): 128175.
[6] Pritchard, Bill, Rodomiro Ortiz, and MeeraShekar, eds. Routledge handbook of food and nutrition security. Routledge, 2016.
[7] Levitt, Barbara, and Clifford Nass. "The lid on the garbage can: Institutional constraints on decision making in the technical core of college-text publishers." Administrative Science Quarterly (1989): 190-207.
[8] Moroney, Laurence, and Laurence Moroney. "An Introduction to Firebase." The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform (2017): 1-24.
[9] Antón-Peset, Adriana, Maria-Angeles Fernandez-Zamudio, and Tatiana Pina. "Promoting food waste reduction at primary schools.A case study." Sustainability 13.2 (2021): 600.
[10] Pingali, Prabhu. "Agricultural mechanization: adoption patterns and economic impact." Handbook of agricultural economics 3 (2007): 2779-2805.
[11] Moroney, Laurence, and Laurence Moroney. "The Firebase RealtimeDatabase." The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform (2017): 51-71.
[12] Kohno, Tadayoshi, et al. "Analysis of an electronic voting system." IEEE Symposium on Security and Privacy, 2004.Proceedings. 2004. IEEE, 2004.
[13] Joorabchi, Mona Erfani, Ali Mesbah, and Philippe Kruchten. "Real challenges in mobile app development."2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement.IEEE, 2013.
[14] Boryan, Claire, et al. "Monitoring US agriculture: the US department of agriculture, national agricultural statistics service, cropland data layer program." Geocarto International 26.5 (2011): 341-358.
[15] Chikae, Miyuki et al. (1979-1985) "Estimation of maturity of compost from food wastes and agro-residues by multiple regression analysis" (2006)