ENHANCING RESTAURANT MANAGEMENT: A USER-FRIENDLY APP FOR IMPROVED SCHEDULING, COMMUNICATION, AND OPERATIONAL EFFICIENCY

Fenghao Liu[1], Jiatong Liu[2], Soroush Mirzaee[3]

[1]Portola High School, 1001 Cadence, Irvine, CA 92618
[2]University of California, Irvine, CA 92697
[3]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Overall, this app was created to solve the challenges of restaurant management [1]. We designed it to be simple and user-friendly for both managers and employees. Key features include secure authentication using Firebase, a weekly scheduling tool for managers, and an emergency contact feature to quickly find available employees [2]. Employees can manage their availability with a week-view calendar. The app's functionalities support check in check out updates for attendance and shift management, improving overall communication and coordination within the restaurant. To test its effectiveness, we surveyed managers and employees who used the app, receiving mostly positive feedback. Future updates will add direct messaging and inventory calculations within the app to incorporate more core features to the app. This app effectively boosts productivity and reduces miscommunication in restaurants, addressing common scheduling challenges and providing an solution. By focusing on enhancing communication and operational efficiency, our app aims to improve the restaurant management experience and efficiency.

KEYWORDS

Restaurant Management, Scheduling Tool, Operational Efficiency, Communication Enhancement

1. INTRODUCTION

The problem that inspired us to create this program is that many of the scheduling apps available today are hard to use. These apps often lack important features needed for managing employees well. Because of this, many restaurants and companies have trouble keeping track of their workers, leading to big issues at work. Without a good scheduling system, managers may find it hard to assign shifts and keep track of attendance, causing confusion and mistakes [3]. This confusion can lead to slower service because employees might not know their scheduled hours or could arrive late or miss shifts entirely. These mistakes can lower the quality of service for customers, hurting the business's reputation and revenue. Also, managers have a harder time keeping everything organized and focused. The lack of good scheduling tools can cause miscommunication between staff and management, making problems at work even worse. In the worst cases, these issues can make employees feel frustrated because of misunderstandings about their schedules. This not only affects the employees but also lowers their peers and increases the

David C. Wyld et al. (Eds): DSCC, AI&FL, SESBC, CSE– 2024
pp. 95-104, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141609

number of people leaving the job. By creating a more user-friendly and feature-supportive scheduling application, our program aims to fix these common problems, ensuring smoother operations, better communication, and improved overall efficiency at work [4].
Our application was created as an effort to allow managers to quickly search for and contact workers who are free to cover for their peers. In order to solve this problem, the application is built to automatically only display employees who state that they are free to work at the time. Another important feature worth noting is the emergency contact feature which differentiates this application from other mobile restaurant competitors. The emergency contact feature allows the manager to quickly contact any of the employees based on their availability and not only by their schedule [5]. Just like other solutions, this app offers normal features such as clocking in and clocking out but also some of the other features that most restaurant applications do not allow on their mobile version. The reason these features were implemented on the mobile application is because most of the restaurant applications are reliant on the desktop view web version to change the schedule or to add employees which is not a solution that everyone would be able to use. Having a standalone application will allow the managers to access most of the features they need on their mobile devices, rather than seeing the need to use a computer will allow for faster changes and more convenience especially during work time.

2. CHALLENGES

One of the sections of the manager’s view of the app is the emergency contact which will allow them to see all of the employees, their roles, and their contact information based on the availability of each employee. Some of the skeptical questions that might arise from this system include:

2.1. Emergency Contact

What is the point of having an emergency contact page based on availability? Especially in bigger restaurants with bigger employee ranges, it is hard for managers to usually find coverage or keep everyone's schedule in mind. This would not only help them to find the employees based on the days that they are available, but it would also makes managing the availability much easier. Why not only have a page to see all of the employees and their contact information like other restaurant applications? After asking a few restaurant managers what features they would like to have in their application that nobody offers, they pointed out that neither of the competitors allows for filtering the employees based on their availability and it is only set by the schedule.

2.2. Authentication

One of the key differences between a restaurant app and most of the other apps is how different the application interface would be for the managers and the employees in every role.

What makes it different? For starters, one of the key differences is that employees cannot register for accounts by themselves and then be added to the application but their account has to be made from the manager’s account and then they will be able to log in. Logging in will also be different as the server will route the user to specific pages depending on their registered role.

If the authentication is complex, how do you ensure the application is user-friendly enough for the majority to use? When the application is first installed, the user will only see the login page or if there is a manager that will need to register their restaurant and their account they will register. Since all of the other users will create their accounts with the manager, the only thing they will need to do on their phone is to login using their email and password.

2.3. Creating Weekly Schedules on A Mobile Application Instead of A Web Application

This was by far one of the most questioned decisions of the application as most of the competition only allows the users to view the schedules from the mobile app but if there is a need for the managers to create or alter the weekly schedule, they will need to make the changes with a laptop or desktop computer.
Isn’t it harder to make the schedules on phones? Technically it is. But not everyone has access to their computer on the go or sometimes it might be more convenient for them to make the modifications needed through a mobile app.

3. SOLUTION

The application is divided into two sections for employees and managers. The managers should be able to create accounts for the employees so the first screen that meets when they open the app is the login page which gives them the option to sign up as a manager. After that the manager will have access to their dashboard where they can see the list of employees and their contact information, the Let's use Instagram as a brief example [6]. When you open Instagram, you have to either make an account or log into an existing account. Then from there, we have our feed, and we are shown posts made by other people. We can view their accounts by clicking on the user icon in those posts, and we can view their other posts and follow them. Or, we could go to another screen in Instagram at the bottom by using the Hotbar. We can view short-form videos or use the built-in messenger.

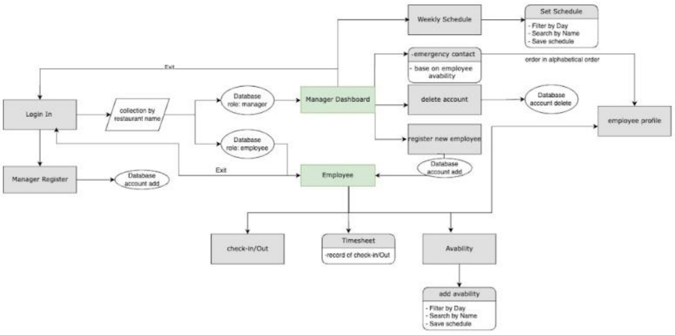
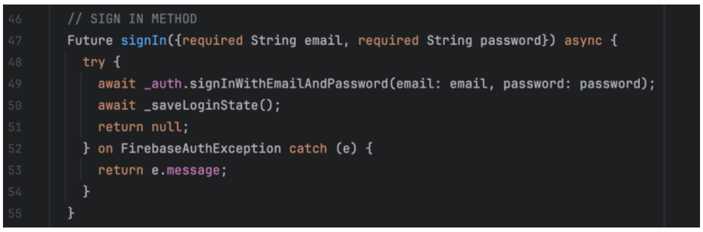


Figure 1. Overview of the solution

The Authentication system uses Firebase for backend services, including secure user authentication via email and password logins [7]. Firebase Authentication differentiates between employee and manager roles, directing users to appropriate interfaces based to their verified status, ensuring role-specific access and enhanced safety. Only managers are able to create accounts for employees, ensuring the control and integrity of user registrations are maintained.



```
// SIGN IN METHOD
Future signIn({required String email, required String password}) async {
  try {
    await _auth.signInWithEmailAndPassword(email: email, password: password);
    await _saveLoginState();
    return null;
  } on FirebaseAuthException catch (e) {
    return e.message;
  }
}
```

Figure 2. Screenshot of code 1

The sign In function is an essential component of a Dart application, designed to authenticate users via Firebase's backend services [8]. It begins by invoking Firebase's sign In With Email And Password method with the user's email and password, using the await keyword to ensure that the application waits for Firebase to respond before proceeding. This method checks the credentials against the database and, upon successful authentication, the next step involves saving the login state with another await call, likely storing session tokens or similar data to maintain the user's logged-in status securely. If the authentication process completes without errors, the function returns null, indicating a successful login. However, if Firebase encounters issues, such as incorrect credentials or network problems, it throws a Fire base Auth Exception. The function catches this exception and returns the error message, providing the user with immediate feedback on what went wrong. This setup not only secures the authentication process but also ensures that user access is appropriately managed based on their roles, enhancing overall application security and functionality.

The Availability Page is designed to help employees manage their availability using a week-view calendar [9]. Integrated with Firebase, this page allows users to view, add, or update their available times based on their schedule. The calendar dynamically updates to reflect changes and interacts with the user for more precise time management. This implementation is crucial for efficient planning and coordination in workplaces that rely on precise scheduling.

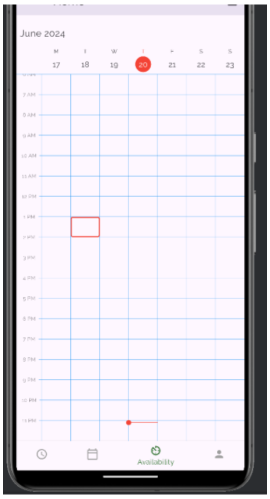
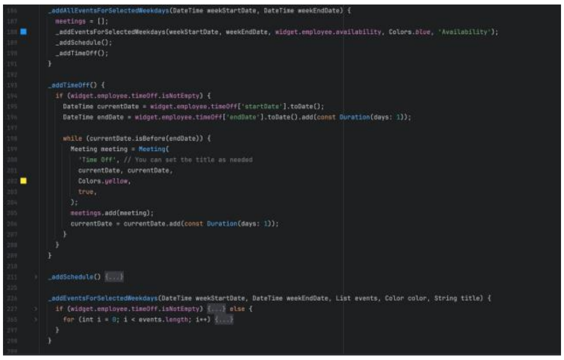


Figure 3. Screenshot of the calendar

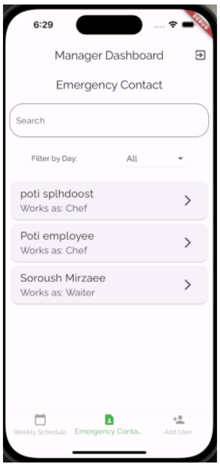


```
_addAllEventsForSelectedWeekdays(DateTime weekStartDate, DateTime weekEndDate) {
  meetings = [];
  _addEventsForSelectedWeekdays(weekStartDate, weekEndDate, widget.employee.availability, Colors.blue, 'Availability');
  _addSchedule();
  _addTimeOff();
}

_addTimeOff() {
  if (widget.employee.timeOff.isNotEmpty) {
    DateTime currentDate = widget.employee.timeOff['startDate'].toDate();
    DateTime endDate = widget.employee.timeOff['endDate'].toDate().add(const Duration(days: 1));

    while (currentDate.isBefore(endDate)) {
      Meeting meeting = Meeting(
        'Time Off', // You can set the title as needed
        currentDate, currentDate,
        Colors.yellow,
        true,
      );
      meetings.add(meeting);
      currentDate = currentDate.add(const Duration(days: 1));
    }
  }
}

_addSchedule() {...}

_addEventsForSelectedWeekdays(DateTime weekStartDate, DateTime weekEndDate, List events, Color color, String title) {
  if (widget.employee.timeOff.isNotEmpty) {...} else {
    for (int i = 0; i < events.length; i++) {...}
  }
}
```

Figure 4. Screenshot of code 2

The methods provided focus on managing and visualizing an employee's schedule within a calendar interface. The function _add All Events For Selected Weekdays serves as a central orchestrator, initially clearing any existing meeting data and then populating the calendar with new entries for availability, scheduled shifts, and time off within the specified week. This is achieved through calls to _add Time Off, _add Schedule, and _add Events For Selected Weekdays. The _add Time Off function iterates over the employee's designated time off, creating meetings highlighted in yellow for each day off. Conversely, _add Schedule scans through fixed work schedules, marking these periods in green to indicate confirmed working times. Lastly, _add Events For Selected Weekdays adds general availability based on the employee's selections, using blue to denote available times. These methods ensure the calendar accurately reflects all aspects of an employee's commitments, enhancing clarity and ease of scheduling adjustments within the application.

The Emergency Contact Page facilitates quick access to employee emergency contacts, integrating with employee availability data. Managers can efficiently identify and communicate with available staff on specific days, enhancing responsiveness in emergency situations. This feature streamlines critical communication within a restaurant management system [10].



Manager Dashboard

Emergency Contact

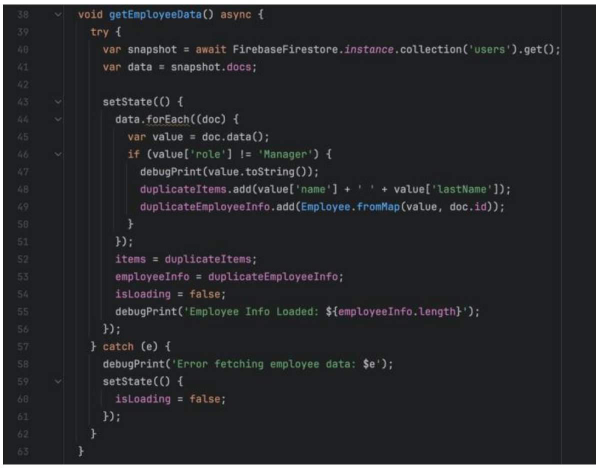
Filter by Day: All

poti splhdoost
Works as: Chef

Poti employee
Works as: Chef

Soroush Mirzaee
Works as: Waiter

Figure 5. Screenshot of emergency contact



```
void getEmployeeData() async {
  try {
    var snapshot = await FirebaseFirestore.instance.collection('users').get();
    var data = snapshot.docs;

    setState(() {
      data.forEach((doc) {
        var value = doc.data();
        if (value['role'] != 'Manager') {
          debugPrint(value.toString());
          duplicateItems.add(value['name'] + ' ' + value['lastName']);
          duplicateEmployeeInfo.add(Employee.fromMap(value, doc.id));
        }
      });
      items = duplicateItems;
      employeeInfo = duplicateEmployeeInfo;
      isLoading = false;
      debugPrint('Employee Info Loaded: ${employeeInfo.length}');
    });
  } catch (e) {
    debugPrint('Error fetching employee data: $e');
    setState(() {
      isLoading = false;
    });
  }
}
```

Figure 6. Screenshot of code 3

The get Employee Data function is an asynchronous method within the Emergency Contact Page designed to fetch and process employee data from a Fire store database. When invoked, it attempts to retrieve all user documents from the 'users' collection. Each document corresponds to an employee, whose data is encapsulated as a document snapshot.

The function filters out documents where the role is labeled as 'Manager', focusing on other employees to populate the emergency contact list. For each relevant employee, their first and last names are concatenated and added to the duplicate items list, which acts as a temporary storage for search functionality. Simultaneously, it constructs an Employee object from the document's data using the Employee. From Map method, which includes the document's ID, and adds this to the duplicate Employee Info list.

Once all appropriate employees are processed, the function updates the items and employee Info lists with data from their respective duplicates. This operation is performed within a set State call to ensure the UI reflects the updated data. If the data fetch fails at any point, an error message is printed, and the is Loading state is set to false to indicate completion or failure of the data loading process. This method ensures that the application has a responsive and up-to-date list of non-managerial employee contacts available for emergency situations.

4. EXPERIMENT

4.1. Experiment 1

To test out how the application would apply to real-world scenarios, we decided to take it to the test by letting some restaurant managers and employees use the app and compare it with what they are using currently. After they spent a bit of time with the application, they were asked some questions in the form of a survey in Google Forms.

Since the application interface is different for employee and manager views, and the functionality of it differs, we decided to ask them questions related to their view. Most of the questions were ranking how a particular feature was or how they would rate the experience but we tried to get feedback for the ease of use and some of the missing components of the application.

1. How frequently do you use our restaurant software app? (Daily, a few times a week, weekly, Rarely)

2. How easy is it to manage employee schedules using our app? 1- 5

3. How helpful is the app in providing real-time updates on employee attendance? 1-5

4. To what extent has the app improved your ability to manage labor costs and scheduling efficiency? 1- 5

5. How likely are you to recommend our employee management features to other restaurant managers? 1-5

6. Have you encountered any issues with the scheduling or attendance features? Y/N

7. If your answer in above question is Yes. Please describe.

8. What training or support do you think is necessary for new users to effectively use the employee management features of the app?

The main reason we chose these questions is that we wanted to ask questions that are more broad and to the point but give the managers the ability to give us some written feedback which will be discussed later.

Moreover here are the questions for the second survey which the employees took.

1. How user-friendly do you find the app's interface?

2. How useful do you find the scheduling feature in managing your shifts?

3. How has the extra features of the app impacted your overall job performance?

4. How has the app influenced your ability to manage your time and tasks?

5. Would you recommend this app to other employees?

6. How satisfied are you with the app overall? (Rating scale: 1-5)

7. How likely are you to continue using this app in the future?

8. From experience, many of the restaurant mobile software are really basic and limit the functionality of the regular employees, Hence is the reason why we questioned them with some of the key features straight and how they would rate them.

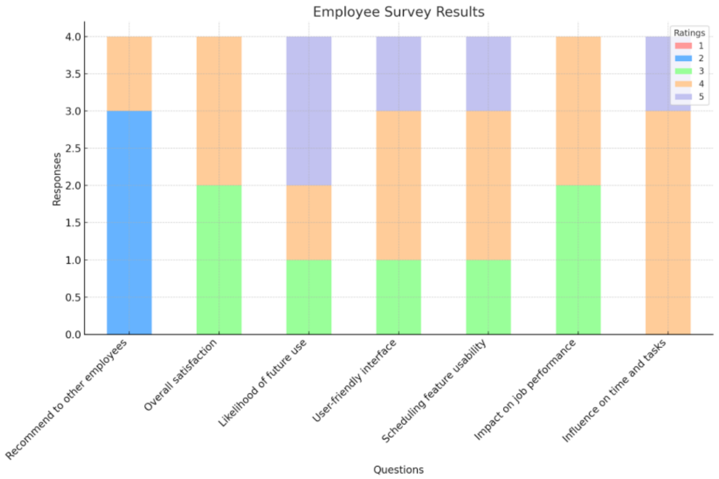


Figure 7. Emloyee survey results

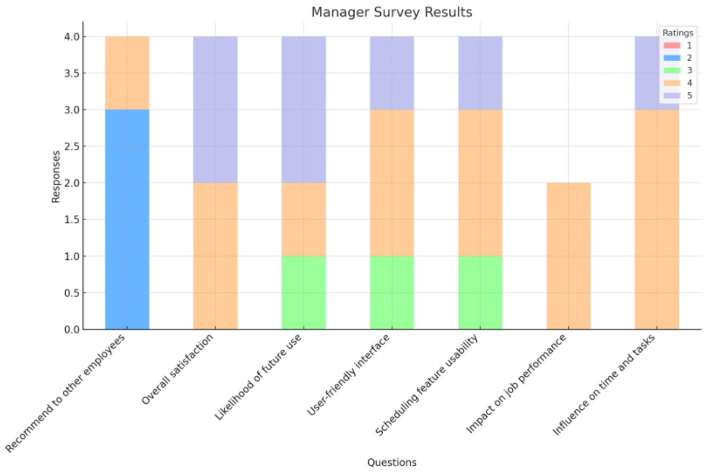


Figure 8. Manager survey results

The average score for the employee on the surveys was 3 and most of the people who took the survey were very moderate with their answers. However, in the employees, we see that there were more people who liked the features and we have more scores of 4 and 5 overall. Moreover, among all of the questions, I think the average was moderate and positive. From both surveys, the most positive responses were “How useful do you find the scheduling feature in managing your shifts?” due to the fact that many restaurant management apps make schedules complicated for the employees, especially with the filters but the simpleness of the schedule was one of the features that got the most compliments. Surprisingly enough, the worst rating out of all of the questions was for “ How helpful is the app in providing real-time updates on employee attendance?” from the manager’s survey. After asking for feedback from the managers, they all expressed how nice it would have been to see the people who are clocked in at the moment in the emergency contact page which can be added as a feature in the future.

5. Related Work

Research Paper: "E-restaurant management system based on Laravel framework" by Ghusoon Idan Arb and Hazim M. Alkargole(Pubs AIP) [11].

Methodology: This research describes the development of an E-Restaurant Management System (ERMS) using the Laravel PHP framework, focusing on improving service efficiency and customer interaction digitally.

Comparison: This system and the DeliDaniSoft app both target restaurant management through digital platforms. However, Deli Dani Soft specifically focuses on employee scheduling and attendance management, which are crucial for operational efficiency but not addressed by the ERMS. By including features such as real-time employee check-ins and check-outs and integrating these with schedule management, it provides a comprehensive solution that not only enhances managerial oversight but also facilitates compliance with labor laws and improves workforce management.

Research Paper: "Mobile Apps Use and WOM in the Food Delivery Sector" by Belanche, Flavián, and Pérez-Rueda (MDPI)[12].

Methodology: Analyzing consumer behavior through online surveys, this study examines the impact of mobile apps on user lifestyle compatibility and security perceptions.

Comparison: This paper provides a close analogy to restaurant management apps by addressing user interaction and security—a core aspect of app development. However, it specifically delves into how user lifestyle and security perceptions influence app usage, which could be pivotal for optimizing user interface and security features in our app. While Belanche et al.'s study emphasizes understanding consumer behavior within the food delivery sector, our app addresses operational efficiency through scheduling and attendance management in restaurants. It could leverage insights from their findings by ensuring the user interface accommodates restaurant employees' lifestyles and enhances security features, which could improve user adoption and satisfaction.

Research Paper: "Mixed-integer programming models for an employee scheduling problem with multiple shifts and work locations" By Salem M.AI – Yakoob & Hanif D.Sherali (MDPI)[13].

Methodology: This study introduces a complex mathematical modeling approach using mixed-integer programming to optimize employee schedules across various shifts and locations. It focuses on maximizing efficiency while meeting a set of constraints like employee availability, labor laws, and operational needs.

Comparison: The mathematical rigor of Al-Yakoob and Sherali's research provides a robust framework for handling complex scheduling scenarios that could be applicable in large, multi-location restaurant chains. In contrast, your app may utilize a simpler, more user-friendly approach tailored specifically for single-location or smaller chain restaurants, focusing on ease of use without requiring deep technical or mathematical expertise. While their model offers scalability and precision for large operations, your app prioritizes immediate, practical usability and quick adjustments to daily restaurant operations, which can be more beneficial for restaurant managers needing to make swift schedule changes.

6. CONCLUSIONS

Our restaurant management app currently allows managers to schedule staff based on availability and includes an emergency contact feature. However, it lacks direct messaging and a complete inventory management system, which are crucial for larger restaurants [14]. Looking ahead, our main focus is on improving these areas. We aim to introduce a messaging tool for real-time communication, making it easier for teams to collaborate and solve problems quickly. This feature will enable managers to communicate efficiently with staff, ensuring smoother shift coordination and faster resolution of issues. Additionally, we plan to develop a comprehensive inventory system within the app. This system will empower managers to monitor stock levels in real-time, streamline ordering processes, and optimize resource allocation based on demand fluctuations. Moreover, integrating advanced analytics for predictive staffing and inventory insights is a priority. This enhancement will enable managers to make informed decisions, improving overall operational efficiency and planning. In the future, our goal is to continually enhance the app to meet evolving industry needs. By focusing on enhancing communication, implementing an up to date inventory management feature, and leveraging data analytics, we aim to provide a comprehensive solution that supports effective restaurant management and enhances the overall dining experience for customers [15].

REFERENCES

[1] Cheong, Soon Nyean, Wei Wing Chiew, and Wen Jiun Yap. "Design and development of Multi-touchable E-restaurant Management System." 2010 International Conference on Science and Social Research (CSSR 2010). IEEE, 2010.

[2] Khawas, Chunnu, and Pritam Shah. "Application of firebase in android app development-a study." International Journal of Computer Applications 179.46 (2018): 49-53.

[3] Carter, Michael W., Gilbert Laporte, and John W. Chinneck. "A general examination scheduling system." Interfaces 24.3 (1994): 109-120.

[4] Craig, Robert T. "Communication as a practice." Communication as… perspectives on theory (2006): 38-49.

[5] Zakariassen, Erik, Robert Anders Burman, and Steinar Hunskaar. "The epidemiology of medical emergency contacts outside hospitals in Norway-a prospective population based study." Scandinavian journal of trauma, resuscitation and emergency medicine 18 (2010): 1-9.

[6] Instagram, POSTING ON. "Instagram." Online unter: https://www. wissenschaftskommunikation. de/format/instagram (2016).

[7] Barkadehi, Mohammadreza Hazhirpasand, et al. "Authentication systems: A literature review and classification." Telematics and Informatics 35.5 (2018): 1491-1511.

[8] Hajslova, Jana, Tomas Cajka, and Lukas Vaclavik. "Challenging applications offered by direct analysis in real time (DART) in food-quality and safety analysis." TrAC Trends in Analytical Chemistry 30.2 (2011): 204-218.

[9] Huang, Dandan, Melanie Tory, and Lyn Bartram. "A field study of on-calendar visualizations." arXiv preprint arXiv:1706.01123 (2017).

[10] Harsin, Jayson. "Post-truth and critical communication studies." Oxford research Encyclopedia of communication. Oxford University Press. https://doi. org/10.1093/acrefore/9780190228613.013 757 (2018).

[11] Arb, Ghusoon Idan, and Hazim M. Alkargole. "E-restaurant management system based on Laravel framework." AIP Conference Proceedings. Vol. 2591. No. 1. AIP Publishing, 2023.

[12] Belanche, Daniel, Marta Flavián, and Alfredo Pérez-Rueda. "Mobile apps use and WOM in the food delivery sector: the role of planned behavior, perceived security and customer lifestyle compatibility." Sustainability 12.10 (2020): 4275.

[13] Al-Yakoob, Salem M., and Hanif D. Sherali. "Mixed-integer programming models for an employee scheduling problem with multiple shifts and work locations." Annals of Operations Research 155 (2007): 119-142.

[14] Maier, Ronald, and Thomas Hadrich. "Knowledge management systems." Encyclopedia of Knowledge Management, Second Edition. IGI Global, 2011. 779-790.

[15] Motiwalla, Luvai, et al. "Leveraging data analytics for behavioral research." Information Systems Frontiers 21 (2019): 735-74.