Research Article

MANAGING CORPORATE CLEANING OPERATIONS WITH A FLUTTER-DEVELOPED MOBILE APP: MAXIMIZING EFFICIENCY WITH THE CLEANER STAFF APP

Authors: Cem Özkurt

To cite to this article: Öztürk, C. MANAGING CORPORATE CLEANING OPERATIONS WITH A FLUTTER-DEVELOPED MOBILE APP: MAXIMIZING EFFICIENCY WITH THE CLEANER STAFF APP, International Journal of Engineering and Innovative Research, 6(3), p 135-150.

DOI: 10.47933/ijeir.1508477

International Journal of Engineering and Innovative Research

http://dergipark.gov.tr/ijeir

MANAGING CORPORATE CLEANING OPERATIONS WITH A FLUTTER-DEVELOPED MOBILE APP: MAXIMIZING EFFICIENCY WITH THE CLEANER STAFF APP

Cem Özkurt*

[1]Sakarya University of Applied Sciences, Faculty of Technology, Department of Computer Engineering, Sakarya, Türkiye.

*Corresponding Author: cemozkurt@subu.edu.tr
(**Received:** 01.07.2024; **Accepted:** 11.08.2024

https://doi.org/10.47933/ijeir.1547634

ABSTRACT: This study introduces a mobile application designed to streamline and supervise institutional operations. The application facilitates seamless coordination for cleaning staff and supervisors. At our university, cleaner staffs used to mark their tasks on paper to prove completion, but this method did not allow for tracking cleaning durations and details. To address these issues, we developed an application that offers a new approach to institutional digital transformation for monitoring cleaning tasks. The application is programmed in Flutter, with a Firebase database, and allows for the recording of task start and completion times, providing managers with a transparent overview. Cleaner staffs access and complete tasks via the user interface, while managers monitor activities, track progress, and generate daily reports through the management interface. A key feature of the application is its ability to archive historical data, allowing managers to access past records and gain insights into operational trends. Additionally, the capability to export reports in PDF format offers accessibility and ease of sharing. The evaluation of the application was conducted by examining data from the Cleaning Services Monitoring Unit under the University's Administrative and Financial Affairs Department. The examined data comprises 280 days and includes 33,587 records. By meticulously recording task start and completion times, the application provides managers with a transparent overview for analysis and evaluation. According to the data from the Cleaning Services Monitoring Unit, the application has increased efficiency by 23% and reduced task completion times by 28%. As a result, cleaner staffs have been able to complete an additional task each day. This demonstrates that the application enhances functionality and makes operations more efficient.

Keywords: Mobile application development, Programming in Flutter, Workflow optimization, Institutional digital transformation, Use of Firebase

1. INTRODUCTION

The mobile application developed in this study addresses the challenges faced by cleaning staff and managers in effectively organizing and supervising cleaning operations. Traditionally, cleaner staffs would mark their tasks on paper, which often led to disputes due to the lack of concrete evidence regarding the duration, location, and completion of tasks. This manual method failed to provide a reliable and transparent record of cleaning activities, resulting in frequent issues between managers and cleaner staffs.

To resolve these issues, the application leverages modern technology to offer a robust and efficient solution. Developed using the Flutter framework, which allows for cross-platform app development, and utilizing Firebase, a comprehensive backend platform by Google, the

application streamlines the process of task assignment, execution, and monitoring. Cleaner staffs receive tasks through the app, record the start and end times of their activities, and update the status upon completion. This real-time data is immediately available to managers, providing a clear and transparent view of ongoing and completed tasks.

The application's key features include:

- **Task Management:** Cleaner staffs are assigned tasks through the app, ensuring they receive instructions in a timely and organized manner.
- **Real-Time Tracking:** Start and end times of tasks are recorded in the database, allowing managers to monitor progress in real-time.
- **Historical Data:** All task-related data is archived, enabling managers to access and review past records for analysis and reporting purposes.
- **Report Generation:** Managers can generate detailed reports on various operational metrics, which can be exported in PDF format for easy sharing and documentation.

The integration of these features ensures a seamless workflow for cleaner staffs and provides managers with the tools needed for effective oversight and planning. By transitioning from paper-based methods to a digital platform, the application enhances transparency, accountability, and efficiency within the cleaning operations.

Overall, the implementation of this application has significantly reduced conflicts and improved satisfaction among both cleaner staffs and managers. The ability to access comprehensive historical data and generate customizable reports empowers administrators to make informed decisions and optimize resource utilization, thereby enhancing the overall efficiency of the cleaning operations.

The application is equipped with significant features such as archiving historical records, which allows managers to gain insights into operational trends and patterns. Furthermore, managers can perform targeted analysis and planning by customizing report parameters based on specific date ranges. Generated reports can be easily exported in PDF format, ensuring accessibility and ease of sharing within the organization.

2. RELATED WORKS

In recent literature, several studies have explored the capabilities and applications of Flutter, an open-source SDK for cross-platform mobile application development. Tashildar et al. (2020) emphasize the importance of cross-platform development and introduce Flutter, highlighting features like Just-in-time and AOT compilation, along with the advantages of Flutter's hot reload for quick experimentation and development [1]. Kuzmin et al. (2020) share their experience using Flutter for developing a mobile application for investment hotels, underscoring Flutter's efficiency in building applications for both Android and iOS from a single codebase [2].

Wu's (2018) thesis compares React Native and Flutter, anticipating their contributions to advancing cross-platform frameworks [3]. Dagne (2019) evaluates Flutter's UI toolkit, comparing it to native and other cross-platform options, exploring its internal architecture, strengths, weaknesses, and competitive potential [4]. Boukhary et al. (2019) propose the Flutter Clean Architecture to address state management issues, providing insights into its design and practical application [5].

Krisnada et al. (2019) discuss the development of a ticket sales mobile application using Flutter, emphasizing its advantages in providing a smooth user experience and a quick development cycle [6]. Mamoun et al. (2021) present the design of a healthcare application prototype using Flutter, showcasing its potential in addressing healthcare challenges through mobile applications [7]. Riyadi et al. (2021) introduce a Flutter-based mobile application for food services, highlighting Flutter's advantages, including super smooth animation [8].

Olsson (2020) compares the performance and aesthetics of Flutter and native applications, exploring scenarios where Flutter might be preferred over native development [9]. Bhagat et al. (2022) review mobile application development based on Flutter, emphasizing its role in reducing the cost and complexity of application creation during the Covid-19 pandemic [10]. Hjort's (2020) thesis evaluates React Native and Flutter for cross-platform development, concluding that React Native produced the best results, but both frameworks are viable options [11].

IŞITAN at el. (2020) compare and evaluate cross-platform mobile application development tools, including Flutter and React Native, based on various parameters [12]. Gonsalves (2019) compares Apache Cordova and Flutter, evaluating the impact on development experience, performance, and perceived quality [13]. Stošović et al. (2022) showcase the use of Flutter in the development of hybrid mobile applications, presenting practical examples [14].

Cheon et al. (2021) explore the conversion of Android native apps to Flutter cross-platform apps, sharing technical experiences and lessons learned during the process [15]. TĂBUȘCĂ et al. (2022) discuss Flutter's capability to develop applications for multiple platforms, emphasizing its use in mobile software applications [16]. Johnson and Wang (2023) utilize Flutter for the mobile application development of gamified automotive reseller team management, providing insights into the development methodology and framework choice [17].

Pop et al. (2021) focus on improving tourists' experiences through an all-in-one touristic mobile app for Timişoara (Romania) using Firebase and Flutter technologies [18]. Allain's (2020) thesis discusses improving productivity and reducing costs of mobile app development with Flutter and Backend-as-a-Service (BaaS), presenting the Alacrity application as a case study [19].

Gałan et al. (2021) present a multi-criteria comparison of mobile applications built with Android and Flutter SDKs, comparing factors such as execution time, CPU usage, and community support [20]. Aminuddin et al. (2022) introduce the Istiqamah App, a mobile application for Sunnah and Hadith reminders using the Flutter framework [21]. Wiriasto et al. (2020) showcase the design and development of an attendance system application using Android-based Flutter [22].

Huynh's (2021) thesis discusses the development of a flashcard mobile application using Flutter, emphasizing the use of Agile methodology in project management [23].

Hamzah et al. (2019) in the realm of mobile application development, several noteworthy contributions have been made to enhance user experiences and address societal needs [24].Introduced an IoT-based smart home system using the Flutter mobile application platform, demonstrating its efficacy in creating a secure environment. Nedyak et al. (2020) delved into cross-platform development tools, highlighting Xamarin's advantages for mobile app creation [26].

Rahman et al. (2020) focused on Flutter framework's versatility, presenting a mobile application for Malay Translated Hadith search engine [27]. Qadir et al. (2020) showcased cross-platform development in a cargo tracking system using Google Flutter technology, catering to the vital transport industry [28]. Additionally, Bisandu et al. (2020) proposed a computer-based test (CBT) mobile application with multimodal biometric authentication to combat examination malpractice in Nigerian tertiary institutions [29].

Gałan et al. (2021) conducted a multi-criteria comparison between Android and Flutter SDK-built mobile applications, emphasizing the speed, efficiency, and community support associated with Android [30]. Wibowo et al. (2021) contributed by designing an agricultural product sales application catalog using the Flutter framework, providing farmers a unified platform for promoting their products [31].

Kleber et al. (2021) directed their efforts towards developing a Flutter-based mobile application, "SOS Syphilis," aimed at mapping data about syphilis infection and curbing its proliferation [32]. Additionally, Krisnada et al. (2020) and Tripathi (2021) presented influential works in this domain [25][33].

Building upon these advancements, recent research has focused on optimizing various aspects of mobile applications. Liaqat et al. (2023) explored the potential of parallelism in reducing execution time for off-loadable parts in applications like face recognition and montage [34]. Kanivets et al. (2023) shared their experiences in using a mobile application for teaching micrometer measurements, leading to improved student success rates [35].

Bi et al. (2023) addressed challenges in mobile edge computing, offering an intelligent computation offloading method for hybrid systems [36]. Zhang et al. (2023) contributed to the optimization of multi-view 3D reconstruction algorithms, aiming for a balance between end-to-end latency and reconstruction quality in a collaborative mobile edge environment [37]. Oh et al. (2023) proposed AppSniffer, a framework for robust mobile app fingerprinting, particularly effective in the presence of virtual private networks (VPNs) [38].

Zaitseva et al. (2023) developed a method to evaluate factors and consequences of mobile application insecurity, emphasizing the importance of identifying and accurately determining factors for reliable forecasting and security assessment [39]. Rezazadeh et al. (2023) provided migration guidelines for services in microservice-based environments, catering to the demand for millisecond-scale response in mobile IoT applications [40].

Furthermore, Lim et al. (2023) introduced SWAM, an integrated memory management technique for mobile devices, addressing the negative effects of existing memory reclamation policies on application responsiveness [41]. Zhang et al. (2023) focused on making SAM mobile-friendly by replacing heavyweight image encoders with lightweight alternatives [42]. Lastly, Huang et al. (2023) proposed CrashTranslator, an approach to automatically reproduce mobile application crashes from stack traces, enhancing software maintenance [43].

3. METHODOLOGY

3.1. Preparation and Implementation of Cleaning Application Observation Form

Data from the University's Cleaning Services Monitoring Unit were used in the organization. The cleaning start and end times for all classrooms and rooms that will be cleaned for 10 months have been developed. This Cleaning Services Monitoring Unit data was applied to 4 buildings

away from the organization's main building, resulting in an average of 7 entries per day with manual methods. A total of 33587 data for 280 days was recorded. The obtained data were determined according to the most appropriate form. Collection and field names that can be transferred to the mobile application.

3.2. Implementation of Qualitative Interview Questionnaire Findings on Mobile Application

The mobile application, developed to monitor and evaluate the cleaning processes within the cleaning services provided by our university, was actively used by the relevant management personnel in the Cleaning Services Monitoring Unit of our university's Administrative and Financial Affairs Directorate. The application was utilized by 50 cleaning staff and 10 managers for 10 months, between February 15, 2023, and December 15, 2023, during weekday business hours. The analysis of the data generated in the NoSQL database of the mobile application during this period, along with the findings related to this questionnaire, are presented in the results section.

3.3. UML (Unified Modeling Language)

The Unified Modeling Language (UML) is a standardized modeling language widely used for designing and analyzing software and systems. UML serves as a tool for understanding, designing, and documenting complex systems. It allows modeling different aspects of systems using various diagram types (such as class diagrams, state diagrams, activity diagrams). UML facilitates communication among stakeholders in the software development process and helps clarify the requirements of the system [44].

3.3.1. UML Diagram of the Cleaner Staff Application

The UML diagram presented in Figure 1 visually represents the basic structure and components of the Cleaner Staff application. This diagram illustrates how different modules and interfaces of the application are related and interact with each other. Firstly, the relationships between user and admin interfaces are clearly depicted in the diagram. While the user interface is used to view cleaning tasks and complete them, the admin interface is utilized to manage tasks and generate reports. The diagram also shows how main collections such as 'Tasks', 'Users', and 'Reports' are organized. These collections store cleaning tasks, user information, and generated reports in the application's central database.

The steps followed by users in managing tasks and generating reports are also indicated in the diagram. Processes such as task acquisition, completion, and report generation are represented through appropriate interfaces. All these components and relationships are important for understanding the functioning of the application and guiding the development process. The UML diagram in Figure 1 serves as a reference point in the design and development process of the Cleaner Staff application, providing a clear specification of the application's requirements and facilitating consensus among stakeholders. This ensures effective development and management of the application.

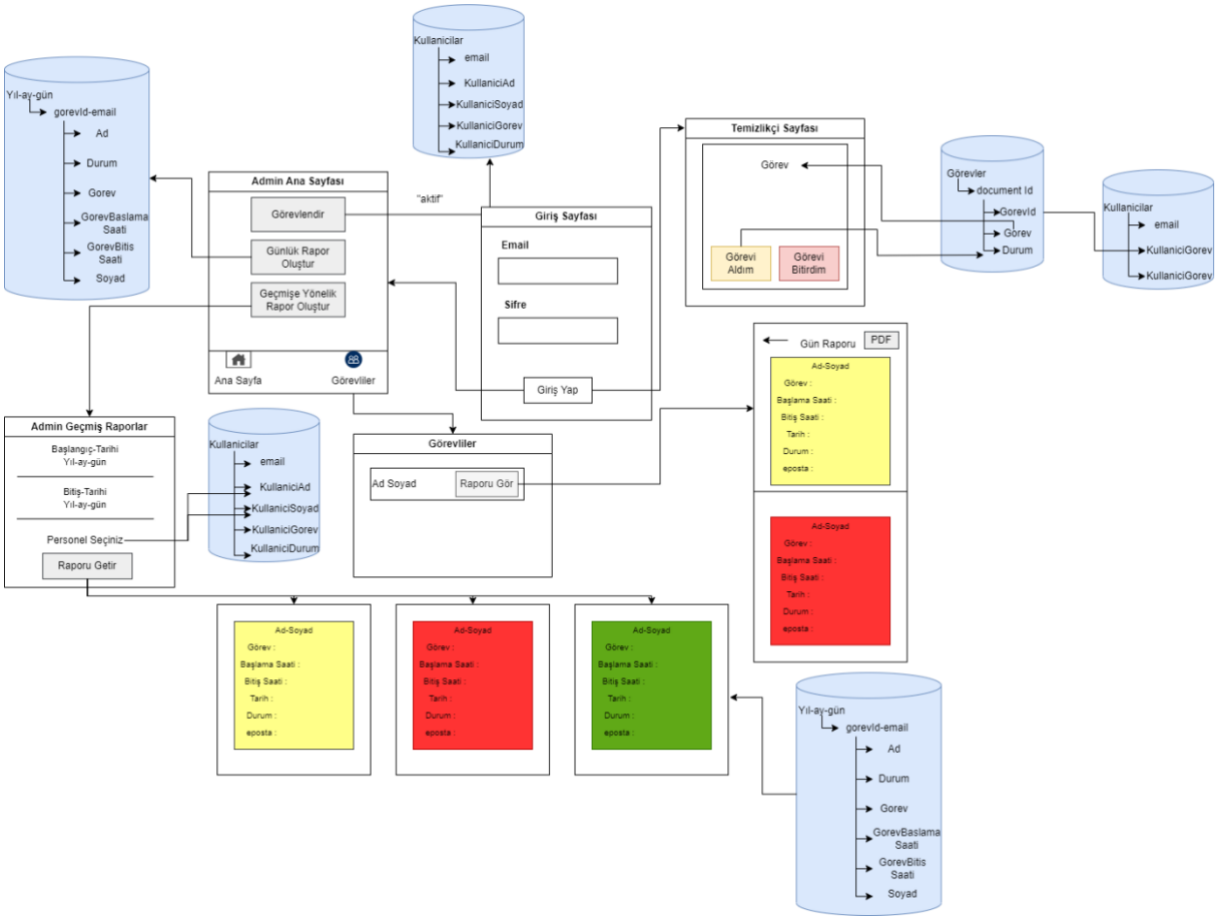

Figure 1. Cleaning Staff Application UML

3.4. Flutter

Flutter is an open-source user interface (UI) software development kit developed by Google. Flutter was used in the development of the Cleaner Staff application. It is designed to create user interfaces for both iOS and Android platforms. However, Flutter also supports application development for the web platform.

Key advantages of using Flutter include a rapid development process, a robust widget library, and platform independence. The rapid development process allows developers to visualize code changes instantly and iterate quickly on the application. The robust widget library enables the creation of consistent and visually appealing user interfaces across "different platforms. As seen in Figure 2, it also boasts a significant advantage of platform independence. It enables developers to develop applications for both iOS and Android devices using a single codebase. Additionally, with Flutter's support for the web platform, applications that can run in web browsers can be created using the same codebase.

The use of Flutter in the development process of the Cleaner Staff application ensured smooth operation across multiple platforms and rapid development. This has helped enhance the user experience of the application and reach a broader audience.

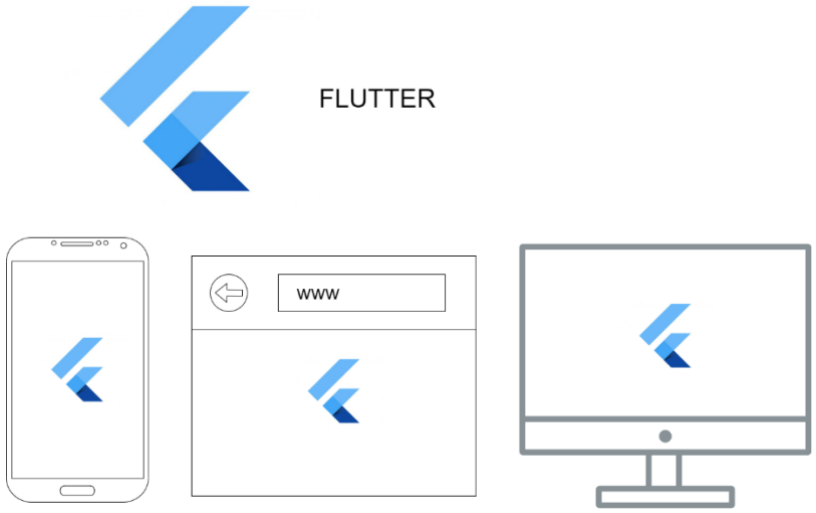

Figure 2. Flutter Platform Independence

3.5. Firebase

Firebase is a platform provided by Google and is a cloud-based mobile and web application development platform. Firebase was used in the development of the Cleaner Staff application. Firebase offers developers a range of tools and services to develop, test, deploy, and monitor their applications.

Key features of Firebase include user authentication, real-time database, storage, server-side logic, analytics, notifications, and testing services. User authentication allows you to authenticate and manage users in your applications. The real-time database enables you to synchronize data in your application in real-time and provides instant updates. The storage service allows users to store, share, and download media files. Server-side logic allows you to perform custom operations in the background of your application. Analytics is an important tool for understanding user behavior and monitoring the performance of your application. Notifications allow you to send customized notifications to users and increase user engagement. Firebase also provides testing services to help improve the quality of your application.

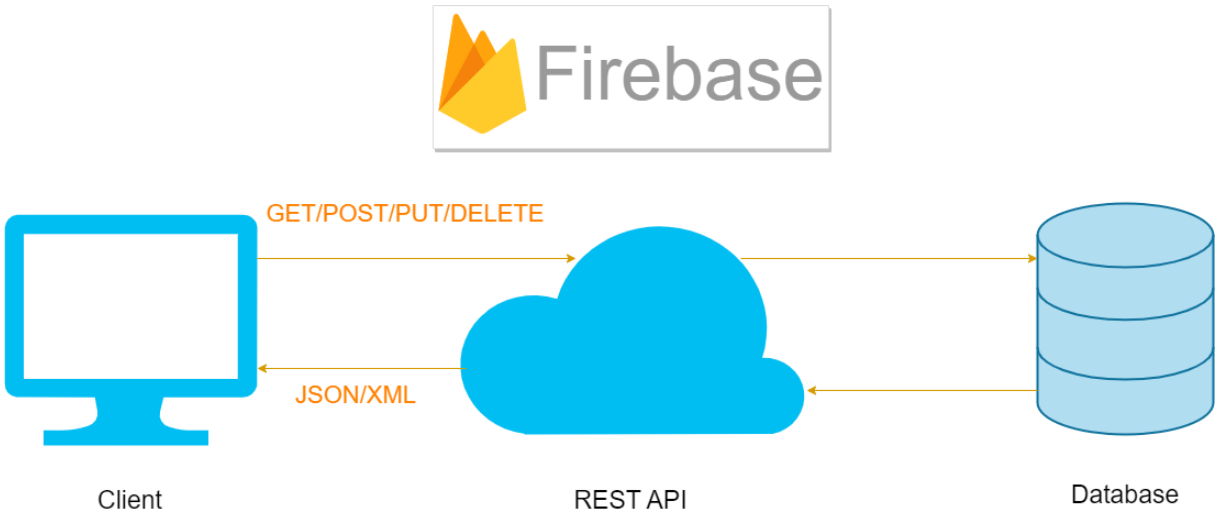

Figure 3. Sending and Retrieving Data with Firebase

The use of Firebase in the development process of the Cleaner Staff application has enhanced the security and effective management of data by leveraging features such as user authentication, real-time database, and storage. Additionally, Firebase's features like analytics and notifications have made it easier to monitor the application's performance and keep users engaged.

With the powerful and comprehensive tools provided by Firebase, the Cleaner Staff application can offer a user-friendly experience while operating securely and efficiently.

3.5.1. FireStore

Firebase Firestore is a real-time cloud-based database service offered by Google. In the development process of the Cleaner Staff application, Firestore was used to store and manage all the application's data. Firestore provides users with fast access to data, offering interactive and up-to-date application experiences.

The application is organized under three main collections: "Görevler", "Kullanıcılar", and "Raporlar". The "Görevler" collection is where cleaning tasks and related information are stored. Each task is represented by a unique document ID, and within this document are the details of the task. The "Kullanıcılar" collection contains user information, with each user represented by a unique document ID such as an email address. User documents include personal information like the user's assigned tasks. The "Raporlar" collection contains reports of completed tasks by users.

In each user's document in the "Kullanıcılar" collection, there is a "KullanıcıGörev" field containing a list of the user's assigned tasks. This list includes an array matching the unique IDs of the user's tasks. These IDs are associated with random document IDs of task documents in the "Görevler" collection.

The statuses of tasks, whether assigned or completed by cleaner staffs, are updated in the "GörevDurumu" field of the relevant task document. When a task is completed, the corresponding task document is added to the "Reports" collection, allowing for the recording of tasks completed by cleaner staffs on that day.

Using Firestore ensures the secure storage and management of application data while also allowing users to access real-time updates. Firestore's flexible and scalable structure allows for customization and expansion according to the application's requirements, enabling efficient operation and meeting users' needs.

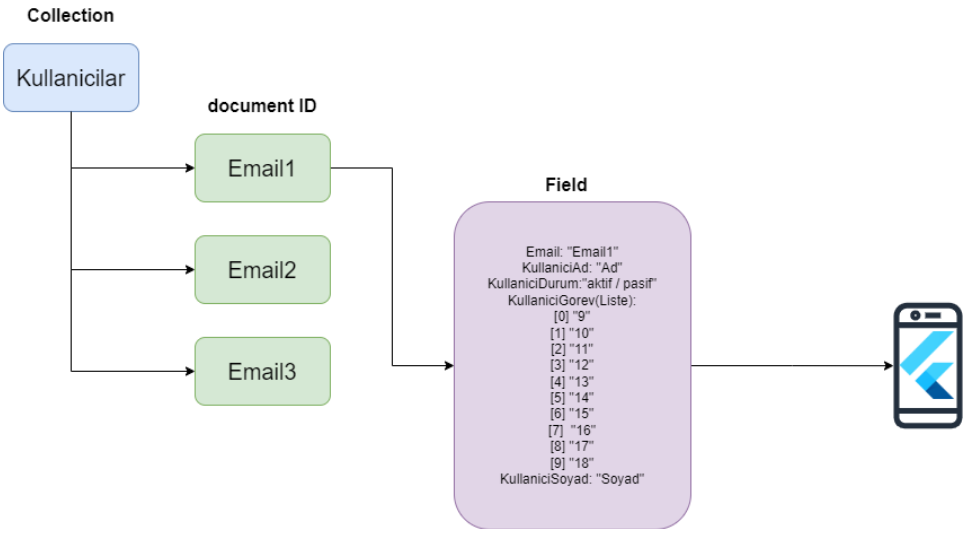

Figure 4. FireStore Kullanici Collection

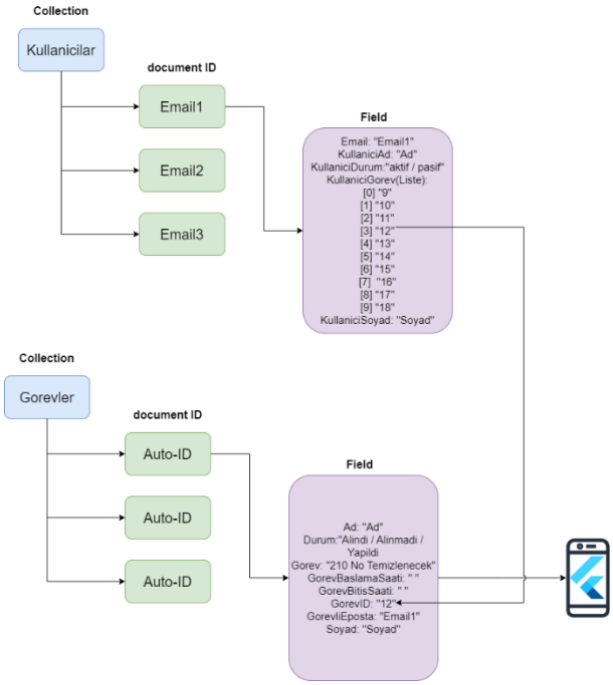

Figure 5. FireStore Gorevler Collection

4. USE OF IN–APP PHOTOGRAPHY

In Figure 6, we present the page visible within the Cleaner Staff application when generating a daily report. This page exclusively displays tasks completed up until the time of generating the daily report. It provides details such as the task assigned to the cleaner staff at that specific hour, the start time, and the completion time of each task. All these details are accessible to the administrator. Additionally, clicking on the PDF icon in the top right corner of this page redirects the user to the page depicted in Figure 8.

Figure 6. Page Where We View Daily Reports

In Figure 7, the depicted page is accessed when an administrator generates a Past Reports and selects the desired date range. Upon selecting the cleaner staff whose activities they wish to review within that timeframe, they are redirected to this page. Here, the administrator can observe which areas the cleaner staff attended to on specific dates and during which time intervals. Cards on this page may appear in three colors: red, yellow, or green. A red card indicates that the assigned task was not completed by the cleaner staff on that day. A yellow card signifies that the task was started but not finished or left incomplete, displaying only the start date and time. Other details are unknown. Meanwhile, a green card indicates that the cleaner staff commenced and completed the task within the same day.

Figure 7. Page Where We View Past Reports

In Figure 8, we present the PDF page that appears when clicking on the PDF icon in the daily report page. This page allows users to obtain a PDF version of the tasks completed on the daily report page. Additionally, it calculates the duration spent on each specified task and displays it to the administrators.

Figure 8. The Daily reports in PDF format

5. OPERATION OF THE APPLICATIONS

The entire operation of the application is on the manager's side. Every morning at 7:30 AM, a notification is sent to remind the manager to start the tasks. If the manager does not start the tasks after this notification, the application automatically starts the tasks. Once the tasks are

started, the first designated task is assigned to the cleaning staff on their page. The cleaning staff also receive a notification that the tasks have started. When the cleaning staff see the assigned task and press the "görevi aldım" button, the task status in the database changes to "görev alındı" and the time they clicked the button is recorded in the database. The cleaning staff then begin the task, and when they complete it and press the "görevi bitirdim" button, the task status in the database changes to "görevi bitirdi" and the time is recorded in the database. They then move on to the next assigned task. This concludes the Cleaning Staff section of the application.

On the manager side, we discussed the task initiation process. Now, let's talk about one of the key advantages of the application: monitoring the work of cleaning staff. The manager's main page has a "Günlük Rapor Oluştur" button, and when clicked, it retrieves all "görev durumu" from the database up to that point in time and presents them to the manager. The manager can then select any cleaning staff to see which tasks they started and completed by that specific time, along with the duration of each task.

The manager's main page also has a "Geçmişe Yönelik Rapor Oluştur" button, which opens a new page when clicked. On this page, the start and end dates need to be selected from the calendar in the year-month-day format. Once selected, the application retrieves all data from the database between those dates and presents it to the manager. The data details include the task date, task start time, task end time, cleaning staff's email, and the cleaning staff's first and last name. This data is displayed in the application using "Card" structures. If the cleaning staff has not completed the task shown on the card, the card appears yellow. If the task has not been taken at all, the card appears red. If the task has been taken and completed, the card appears green.

6. RESULTS

The developed mobile application is a comprehensive solution designed to effectively organize and supervise cleaning tasks. Targeting both cleaner staff performing the tasks and the supervisors overseeing them, the application aims to enhance communication and collaboration between them. With its user-friendly interface, cleaner staff can easily access, perform, and complete tasks, while supervisors can monitor cleaner staff activities, track progress, and manage overall operations.

The application allows cleaner staff to accurately record task initiation and completion times, enabling supervisors to better understand trends and patterns in operational processes. Additionally, the app has the capability to archive historical data, providing supervisors with valuable insights into past records and enhancing operational efficiency.

Table 1. Comparison Table: Average Cleaning Times Measured with Registry and Mobile Application

Cleaning Area	Registry	Mobile App
207	43	25
208	33	36
305	49	38
307	51	35
328	46	30
Fire Stairs	51	48

$$\bar{T}_{Registry} = \frac{\sum_{i=1}^{n} T_{Registry,i}}{n} \quad (1)$$

$$\bar{T}_{Mobile} = \frac{\sum_{i=1}^{n} T_{Mobile,i}}{n} \quad (2)$$

$$\mathrm{S} = \bar{T}_{Registry} - \bar{T}_{Mobile} \quad (3)$$

$$\mathrm{P}_S = \left(\frac{S}{\bar{T}_{Registry}}\right) x100 \quad (4)$$

$$\mathrm{P}_r = \left(\frac{\bar{T}_{Registry} - \bar{T}_{Mobile}}{\bar{T}_{Registry}}\right) x100 \quad (5)$$

- $\bar{T}_{Registry}$: Average cleaning time using the registry data.
- $\bar{T}_{Mobile}$: Average cleaning time using the mobile app data.
- $T_{Registry,i}$: Individual cleaning time from the registry data.
- $T_{Mobile,i}$: Individual cleaning time from the mobile app data.
- n : Number of data points.
- S: Time savings achieved.
- P_S: Percentage of time savings.
- P_r: Productivity increase percentage.

This table and the mathematical expressions below it compare the cleaning times of 5 locations within a university's record book before and after using a mobile application. The goal is to determine time savings and efficiency.
Firstly, $\bar{T}_{Registry}$ and $\bar{T}_{Mobile}$ represent the average cleaning times measured using the registry data and the mobile app data, respectively. These values are calculated by summing the individual cleaning times and dividing by the number of measurements (Equations 1 and 2).
S represent the difference between the registry and mobile app cleaning times, calculated as Equation 3.
P_S is the percentage of time savings, calculated as Equation 4.
P_r represents the percentage increase in efficiency and is calculated using Equation 5.
These equations are used to determine how much time is saved and how efficiency is improved by using the mobile application for cleaning processes. Cleaning times and the reductions in these times are compared based on data from both the registry and the mobile app.

Our statistics were obtained using data from the Cleaning Monitoring Registry of the Administrative and Financial Affairs Directorate of our university. Compared to the data obtained, the application resulted in a 23% increase in the efficiency of cleaner staff and a 28% reduction in the time spent on tasks. As seen in Table 1, according to the registry, Classroom 207 was cleaned on average in 43 minutes before the application usage and increased supervision, but decreased to 25 minutes afterward. Similarly, the average cleaning time for Classroom 305 reduced from 49 minutes to 38 minutes after the application was implemented and supervision level increased. These findings indicate that the application has successfully improved operational efficiency and increased user satisfaction. Particularly, the reporting and analytics features provide supervisors with essential tools for better managing and optimizing operational processes.

In conclusion, the developed mobile application effectively empowers corporate cleaning operations by enhancing collaboration and optimizing operational processes, thus providing a valuable tool for increasing efficiency.

7. CONCLUSION AND DISCUSSION

Our comprehensive mobile application has marked a significant turning point in university cleaning operations. It has brought about notable advancements in key areas such as efficiency, transparency, and collaboration. The data recording and monitoring capabilities have provided significant advantages to both cleaner staffs and supervisors. While offering a user-friendly interface that simplifies tasks for cleaner staffs, it has also enhanced supervision and management capabilities for supervisors.

Furthermore, the archival feature of the application has empowered administrators with deep insights into operational trends and patterns, strengthening decision-making processes and optimizing resource utilization. The reporting and analytical features have streamlined targeted analysis and planning through customizable reports, fostering accessibility and collaboration.

Survey results have highlighted substantial improvements in user satisfaction and operational efficiency, indicating successful resolution of issues between cleaner staffs and supervisors. Overall, the application has fostered a culture of transparency and accountability within the organization. With continuous innovation and adaptation, its effectiveness can be further enhanced, leading to even greater organizational success.

REFERENCES

[1] Tashildar, A., Shah, N., Gala, R., Giri, T., & Chavhan, P. (2020). Application development using flutter. International Research Journal of Modernization in Engineering Technology and Science, 2(8), 1262-1266.

[2] Kuzmin, N., Ignatiev, K., & Grafov, D. (2020). Experience of developing a mobile application using flutter. In Information Science and Applications: ICISA 2019 (pp. 571-575). Springer Singapore.

[3] Wu, W. (2018). React Native vs Flutter, Cross-platforms mobile application frameworks.

[4] Dagne, L. (2019). Flutter for cross-platform App and SDK development.

[5] Boukhary, S., & Colmenares, E. (2019, December). A clean approach to flutter development through the flutter clean architecture package. In 2019 international conference on computational science and computational intelligence (CSCI) (pp. 1115-1120). IEEE.

[6] Krisnada, F. E., & Tanone, R. (2019). Aplikasi Penjualan Tiket Kelas Pelatihan Berbasis Mobile menggunakan Flutter. Jurnal Teknik Informatika Dan Sistem Informasi, 5(3).

[7] Mamoun, R., Nasor, M., & Abulikailik, S. H. (2021, February). Design and development of mobile healthcare application prototype using flutter. In 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE) (pp. 1-6). IEEE.

[8] Riyadi, S., & Cahyono, T. (2021, February). Information system for providing food services based on mobile application using flutter framework. In 4th International Conference on Sustainable Innovation 2020–Technology, Engineering and Agriculture (ICoSITEA 2020) (pp. 164-169). Atlantis Press.

[9] Olsson, M. (2020). A Comparison of Performance and Looks Between Flutter and Native Applications: When to prefer Flutter over native in mobile application development.

[10] Bhagat, S. A. (2022). Review on Mobile Application Development Based on Flutter Platform. International Journal for Research in Applied Science and Engineering Technology, 10(1), 803-809.

[11] Hjort, E. (2020). Evaluation of React Native and Flutter for cross-platform mobile application development.

[12] Işıtan, M., & Koklu, M. (2020). Comparison and evaluation of cross platform mobile application development tools. International Journal of Applied Mathematics Electronics and Computers, 8(4), 273-281.

[13] Gonsalves, M. (2019). Evaluating the mobile development frameworks Apache Cordova and Flutter and

their impact on the development process and application characteristics.
[14] Stošović, S., Stefanović, D., Bogdanović, M., & Vukotić, N. (2022). THE USE OF THE FLUTTER FRAMEWORK IN THE DEVELOPMENT PROCESS OF HYBRID MOBILE APPLICATIONS. KNOWLEDGE-International Journal, 54(3), 477-483.
[15] Cheon, Y., & Chavez, C. (2021, December). Converting Android native apps to Flutter cross-platform apps. In 2021 International conference on computational science and computational intelligence (CSCI) (pp. 1898-1904). IEEE.
[16] TĂBUȘCĂ, A., Coculescu, C., & Pîrnău, M. (2022). FLUTTER TECHNOLOGY AND MOBILE SOFTWARE APPLICATIONS. Journal of Information Systems & Operations Management, 16(2).
[17] Johnson, E., & Wang, G. (2023). Flutter Framework Mobile Application Development of Gamified Automotive Reseller Team Management. Jurnal Info Sains: Informatika dan Sains, 13(03), 1125-1131.
[18] Pop, M. D., & Stoia, A. R. (2021, August). Improving the Tourists Experiences: Application of Firebase and Flutter Technologies in Mobile Applications Development Process. In 2021 International Conference Engineering Technologies and Computer Science (EnT) (pp. 146-151). IEEE.
[19] Allain, H. (2020). Improving productivity and reducing costs of mobile app development with Flutter and Backend-as-a-Service (Master's thesis).
[20] Gałan, D., Fisz, K., & Kopniak, P. (2021). A multi-criteria comparison of mobile applications built with the use of Android and Flutter Software Development Kits. Journal of Computer Sciences Institute, 19, 107-113.
[21] Aminuddin, R., Khan, N. A. M. N., Noor, M. H. M., Ismail, I., Isa, M. A. M., Elias, S. S. M., ... & Ilias, N. F. (2022, May). Istiqamah App: A Mobile Application for Sunnah and Hadith reminder using Flutter framework. In 2022 IEEE 18th International Colloquium on Signal Processing & Applications (CSPA) (pp. 237-242). IEEE.
[22] Wiriasto, G. W., Aji, R. W. S., & Budiman, D. F. (2020, October). Design and development of attendance system application using android-based flutter. In 2020 Third International Conference on Vocational Education and Electrical Engineering (ICVEE) (pp. 1-6). IEEE.
[23] Huynh, T. (2021). A flashcard mobile application development with Flutter.
[24] Hamzah, N. A. B. A., Saad, M. R. B. A., Ismail, W. Z. B. W., Bhunaeswari, T., & Abd Rahman, N. Z. B. (2019). Development of A Prototype of An IoT Based Smart Home with Security System Flutter Mobile. Journal of Engineering Technology and Applied Physics, 1(2), 34-41.
[25] Krisnada, F. E., & Tanone, R. (2019). Aplikasi Penjualan Tiket Kelas Pelatihan Berbasis Mobile menggunakan Flutter. Jurnal Teknik Informatika Dan Sistem Informasi, 5(3).
[26] Nedyak A, Rudzeyt O, Zainetdinov A, Ragulin P, "Mobile Cross-platform App Development Tools", 2020.
[27] Abd Rahman, N., Syamil, F. I. M. R., & bin Rodzman, S. B. (2020). Development of mobile application for Malay translated hadith search engine. Indonesian Journal of Electrical Engineering and Computer Science, 20(2), 932-938.
[28] Qadir, A. M., & Cooper, P. (2020, June). GPS-based mobile cross-platform Cargo tracking system with web-based application. In 2020 8th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-7). IEEE.
[29] Desmond, A. (2020). Farouk (2020). Multimodal Biometric Authentication For a Computer-Based Test (CBT) Application. IRJCS:: International Research Journal of Computer Science, Volume VII, Issue VII, 179-196.
[30] Gałan, D., Fisz, K., & Kopniak, P. (2021). A multi-criteria comparison of mobile applications built with the use of Android and Flutter Software Development Kits. Journal of Computer Sciences Institute, 19, 107-113.
[31] Wibowo, A., Delima, R., & Rachmat, C. A. (2021, February). Designing and Developing an Agricultural Product Sales Application Catalog with a Hybrid Application Development Framework. In IOP Conference Series: Materials Science and Engineering (Vol. 1077, No. 1, p. 012050). IOP Publishing.
[32] Kleber, G., Alves, D., de Paiva, J., Diógenes, D., Valentim, R., & Medeiros, A. (2021). SOS Syphilis: smartphone application for the mapping of syphilis attention networks. Procedia Computer Science, 181, 434-441.
[33] Sahu, S. K., Sharma, M. L., & Tripathi, K. C. (2021). COVID-19 india tracker application using flutter. Int J Mod Trends Sci Technol, 7(1), 10-15.
[34] Liaqat, A., Ilyas, S., & Mushtaq, G. (2023). Distributed Computation Offloading of an application from mobile/IoT device to cloud. arXiv preprint arXiv:2302.02481.
[35] Kanivets, O. V., Kanivets, I. M., Gorda, T. M., Gorbenko, O. V., & Kelemesh, A. O. (2023, March). Using a mobile application to teach students to measure with a micrometer during remote laboratory work. In CTE Workshop Proceedings (Vol. 10, pp. 294-314).
[36] Bi, J., Zhang, K., Yuan, H., & Zhang, J. (2022). Energy-efficient computation offloading for static and

dynamic applications in hybrid mobile edge cloud system. IEEE Transactions on Sustainable Computing, 8(2), 232-244.

[37] Zhang, X., Li, M., Hilton, A., Pal, A., Dey, S., & Debroy, S. (2022, August). End-to-end latency optimization of multi-view 3D reconstruction for disaster response. In 2022 10th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud) (pp. 17-24). IEEE.

[38] Oh, S., Lee, M., Lee, H., Bertino, E., & Kim, H. (2023, April). Appsniffer: Towards robust mobile app fingerprinting against VPN. In Proceedings of the ACM Web Conference 2023 (pp. 2318-2328).

[39] Zaitseva, E., Hovorushchenko, T., Pavlova, O., & Voichur, Y. (2023). Identifying the Mutual Correlations and Evaluating the Weights of Factors and Consequences of Mobile Application Insecurity. Systems, 11(5), 242.

[40] Rezazadeh, A., & Lutfiyya, H. (2023). Multi-microservice migration modelling, comparison, and potential in 5G/6G mobile edge computing: A non-average parameter values approach. IEEE Access.

[41] Lim, G., Kang, D., Ham, M., & Eom, Y. I. (2023, July). SWAM: Revisiting Swap and OOMK for Improving Application Responsiveness on Mobile Devices. In Proceedings of the 29th Annual International Conference on Mobile Computing and Networking (pp. 1-15).

[42] Zhang, C., Han, D., Qiao, Y., Kim, J. U., Bae, S. H., Lee, S., & Hong, C. S. (2023). Faster segment anything: Towards lightweight sam for mobile applications. arXiv preprint arXiv:2306.14289.

[43] Huang, Y., Wang, J., Liu, Z., Wang, Y., Wang, S., Chen, C., ... & Wang, Q. (2024, February). Crashtranslator: Automatically reproducing mobile application crashes directly from stack trace. In Proceedings of the 46th IEEE/ACM International Conference on Software Engineering (pp. 1-13).

[44] Booch, G., Rumbaugh, J., & Jacobsen, I. (1997). UML: unified modeling language. Versão.