

Development of Employee Attendance Management System using Flutter

Muhamad Faiz Akmal Mohamad Noor¹, Alif Faisal Ibrahim^{2*}, Muhammad Nabil Fikri Jamaluddin³
^{1,2,3} College of Computing, Informatics, and Mathematics, Universiti Teknologi MARA Perlis Branch, Arau Campus, 02600 Arau, Perlis, Malaysia.

Corresponding author: * aliffaisal@uitm.edu.my

Received Date: 6 June 2023

Accepted Date: 5 July 2023

Revised Date: 10 August 2023

Published Date: 1 September 2023

HIGHLIGHTS

- The concept of working hours is frequently linked to the operations of businesses and their corresponding levels of productivity.
- The utilisation of conventional approaches to track employee attendance is no longer sufficient to achieve the present operational objectives of businesses.
- The implementation of a contemporary and advanced employee attendance system may be a feasible choice for enterprises of considerable scale, although it may not be suitable for small firms, startups, or non-profit organisations.
- Using Flutter development framework to develop a cross platform attendance management application can provide small businesses with low cost yet reliable solutions to manage their employees.

ABSTRACT

The workforce constitutes a fundamental element of organisations, and effectively addressing the challenges associated with governing employees is crucial for maintaining the overall stability of the firm. In the past, businesses and organisations used to depend on manual attendance tracking systems within their human resources departments to manage their staff. In the contemporary era characterised by significant advancements in technology, several recommendations are put out to enhance the existing system. Nevertheless, it is possible that small enterprises may encounter difficulties in coping with the escalating living expenses and the expenses associated with implementing an advanced attendance management system. Hence, this study proposes the creation of a straightforward yet dependable mobile application for the purpose of documenting and overseeing staff attendance through the utilisation of Flutter. The application was developed in a series of incremental steps utilising the Waterfall methodology. The effectiveness and acceptance of the application were evaluated using a comprehensive testing process that included both functionality and usability testing. Both testing findings yielded favourable outcomes, indicating that the application was favourably received by the responders. The application is anticipated to provide assistance to small enterprises or organisations that have encountered challenges in managing their operating expenses.

Keywords: attendance system, usability testing, user acceptance, mobile development, flutter, dart



INTRODUCTION

The workforce is an essential component of any companies or businesses, and properly managing any issues related to employee governance is vital for ensuring their general stability. Employee attendance is crucial to productivity and to maintain efficiency of a corporate function. For instance, in the context of a project that involves numerous participants, the absence of a crucial team member could potentially result in a delay in project delivery, hence posing a risk to the organization's reputation. Moreover, the absence of a comprehensive attendance management system necessitates the allocation of additional personnel within the human resource (HR) department. Conventional methods of employee attendance tracking, such as the utilisation of punch cards and paper sign-in sheets, can prove to be laborious and inefficient for both employees and human resources personnel, while also being susceptible to human errors. To record their attendance, employees are required to physically locate the time management system device and wait in a queue (Iio, 2016). Furthermore, instances have been documented where dishonest personnel engaged in tampering with of the device. Managing multiple remote working locations or accommodating diverse schedules can provide significant challenges for businesses.

Overall, traditional employee attendance systems are inefficient and can lead to inaccurate attendance records, which can have negative impacts on payroll and compliance with labor laws (G. Kaur, 2018). To track employee attendance, corporations employ a diverse array of methods, encompassing traditional punch cards and spreadsheets, as well as technologically advanced attendance management systems such as biometric devices, RFID, and NFC. However, current operational objectives of firms necessitate a departure from traditional methods of monitoring staff attendance, as they are no longer deemed adequate. One of the primary drawbacks associated with employing a manual-based recording method is the lack of dependability exhibited by employees. In contrast, the use of sophisticated techniques like biometrics or smart devices poses significant challenges, particularly for small enterprises or non-profit entities, mostly attributable to their constrained financial resources (A. Kaur et al., 2022). There are certain restrictions associated with the biometric technology, for instance. The fingerprint attendance method is afflicted by the issue of physical touch between individuals, hence increasing the risk of potential disease transmission (Oo et al., 2018). In addition, it is important to consider the increasing expenses associated with maintaining a standard of living.

LITERATURE REVIEW

Attendance Management

Attendance management refers to the systematic approach employed by organisations to effectively monitor and regulate employee attendance. This practise aims to mitigate potential productivity losses by meticulously tracking employees' working hours, as well as documenting instances of absence and leave taken by employees. Both Ahmed et al., (2019) and Muhtahir O. et al., (2013) highlight the importance of attendance management systems including daily monitoring of staff, computation of overtime, and transfer of pertinent information to the payroll system, as well as manpower analysis. Furthermore, ensuring the accuracy of employees' attendance data is crucial for the effective maintenance of employee monitoring and discipline (Oo et al., 2018). A paper by Ahmed and others (2019) also underscores the significance of maintaining precise attendance records of employees to detect issues within the working environment.



In contemporary society, smartphones have become indispensable devices for individuals, while the availability of affordable Internet data rates further enhances their potential as optimal instruments for documenting employee attendance. The development of applications or apps for several platforms typically necessitates a comprehensive acquisition of developers, hence resulting in increased expenditures. Therefore, small enterprises or organisations have the option to choose a cross-platform application development framework, such as Flutter.

Flutter

It is expected that mobile applications would offer an enhanced user experience in a prompt, effective, and efficient manner. Nonetheless, developers of mobile applications are consistently faced with the challenge of striking a balance between the quantity of code lines, the level of functionality, and the allocation of resources. Additionally, they must contend with the diverse range of platforms and application programming interfaces (APIs) that necessitate their support. Flutter is a software development toolkit for user interface (UI) that has been made open source by Google. This technology enables software developers to construct apps that are natively compiled for several platforms, including mobile, web, and desktop, using a unified codebase.

One of the foremost advantages offered by Flutter is its cross-platform compatibility, enabling developers to create code once and afterwards publish applications across several platforms such as iOS, Android, web, and desktop. This leads to a decrease in the duration and the effort required for development. Flutter applications are compiled in a native manner, hence exhibiting exceptional performance that closely rivals that of applications developed specifically for a certain platform. Furthermore, Flutter boasts a robust and engaged developer community that actively contributes to its ongoing development and enhancement. The community offers a range of materials, tutorials, plugins, and solutions to address prevalent issues. Ultimately, the capacity to address numerous platforms through a unified codebase can result in cost efficiencies in terms of development, maintenance, and upgrades. This is because instead of constructing and managing several distinct applications, a single application can be built and maintained. The Dart programming language is predominantly linked with the Flutter framework.

Dart

The Dart programming language was developed by Google. The framework is specifically engineered to facilitate the development of current online and mobile apps. The major aim of developing Dart was to improve the development process for generating high-performance cross-platform applications (Dart Overview, n.d.). As a consequence, the application user interface (UI) exhibits a high degree of adaptability and responsiveness when displayed on different screen types, aspect ratios, and orientations.

Firestore

Firestore is a complete platform offered by Google that facilitates the development and administration of web and mobile apps. The platform provides a comprehensive range of services including multiple facets of application development, such as backend services, cloud storage, authentication, real-time databases, analytics, and additional functionalities. Firestore is specifically engineered to streamline the process of software development, enabling developers to concentrate on crafting exceptional user experiences without being burdened by intricate infrastructure-related responsibilities. The integration of Firestore with various Google services and its user-friendly nature renders it as a compelling option, particularly for startups, small teams, and projects with constrained resources.



METHODOLOGY

The Waterfall model is a conventional software development technique characterised by its adherence to a linear and sequential approach in the development process. The model is comprised of a sequence of discrete phases, wherein the completion of each phase is a prerequisite for the commencement of the subsequent phase. Figure 1 illustrates the comprehensive breakdown of the many processes required to successfully finish the application process. The methodology process comprises six discrete steps, namely initiation, planning, designing, developing, testing, and documentation.

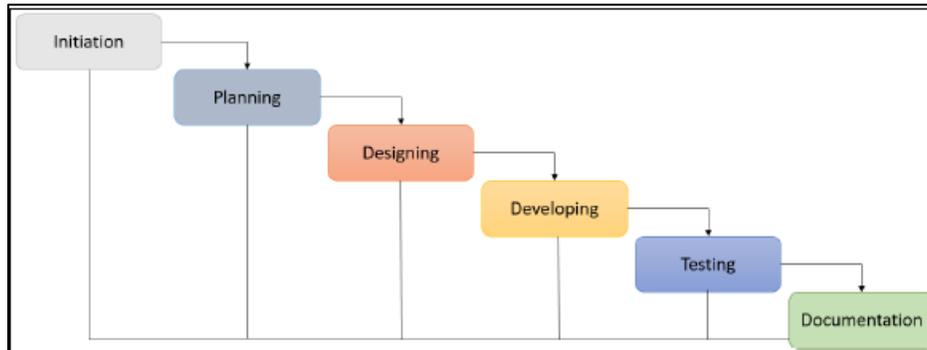


Figure 1: Methodology

The initiation phase is associated with gathering all related information about the project. Furthermore, in this phase, a comprehensive investigation was conducted on several aspects of the project, including the project title, background information, issue statement, objectives, scope, and the significance of the research. During this phase, the researcher engages in a comprehensive review of pertinent academic journal articles and previous research studies. The planning phase of this project is designated as the second phase. This phase encompasses a variety of activities, including the collection of all project-related requirements and the acquisition of necessary resources. The researcher determined the necessary hardware and software prerequisites by considering prior activities. It is necessary to thoroughly analyse each requirement in order to advance to the subsequent phase. The design phase constitutes the third stage of the project. The objective of this phase is to generate a comprehensive plan that fulfils all the specified requirements of the system. Currently, the storyboard serves the purpose of offering a preliminary outline of the project's progression. The selection of the platform is contingent upon the comprehensive identification of all requisite inputs, outputs, and procedures. Moreover, a user interface refers to a meticulously crafted design that incorporates all the necessary technology

The fourth phase will encompass the development stage. In this phase, the researcher constructs the application in accordance with the needs and designs that have been obtained and evaluated in preceding phases. Throughout the process of development, various reliable tools and technologies, such as Dart, were utilised. Furthermore, this phase encompasses activities related to the completion of users' interfaces and the establishment of a connection to Firebase for the purpose of storing attendance records. Once the development phase has concluded, it is imperative to conduct thorough testing of the application to validate the proper functioning of all its components and ensure the expected outcomes are being generated. Moreover, the process of testing plays a critical role in verifying that the application meets all functional and non-functional requirements. The testing step was categorised into two distinct areas, namely functionality testing and usability testing. The researcher conducts a functionality test to assess the system



as a whole, identifying any potential faults and ensuring that the application is functioning correctly. The usability test was administered to target users to assess the users' acceptance of the application.

Ultimately, the documentation step will serve as the final stage of the process. All relevant data, including testing results, is collected and analysed in order to gain a deeper understanding and provide additional clarification. The anticipated result of this phase would be the production of a comprehensive report.

DEVELOPMENT

The flowchart illustrates the sequential process of navigating through the application. Figure 2 depicts the flow chart representing the system. The clock in and clock out information of users is maintained in Firebase.

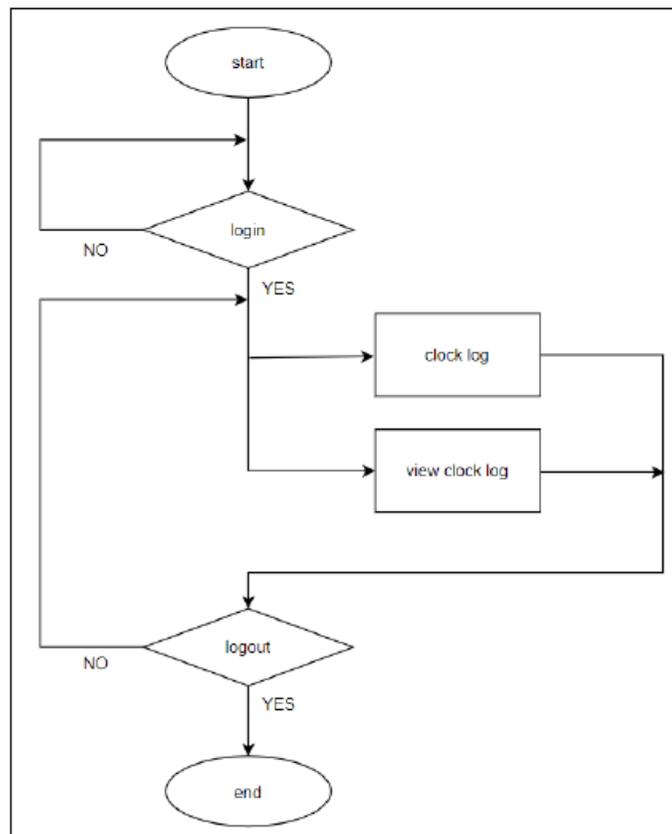


Figure 2: Flow chart for Employee Attendance Management System

Figure 3 shows examples of interface for user login and the clocking page, which is the main interface for the application. Prior to accessing the application's content, users are required to authenticate themselves by logging in using secure credentials. After the process of authentication is completed successfully, users are provided with the option to select between two sections, which are visually represented by two tabs located below the screen. One of the primary components of the application is the main Clocking Page, while another integral portion is the Clock Log History. For the clocking page, in the absence of attendance



records for the present date, both clock-in and clock-out data will be unavailable. Users will have the capability to accurately record their clock-in information by effortlessly sliding the button labelled "Slide to clock-in". Simultaneously, the geographical coordinates of the user's location are also logged.

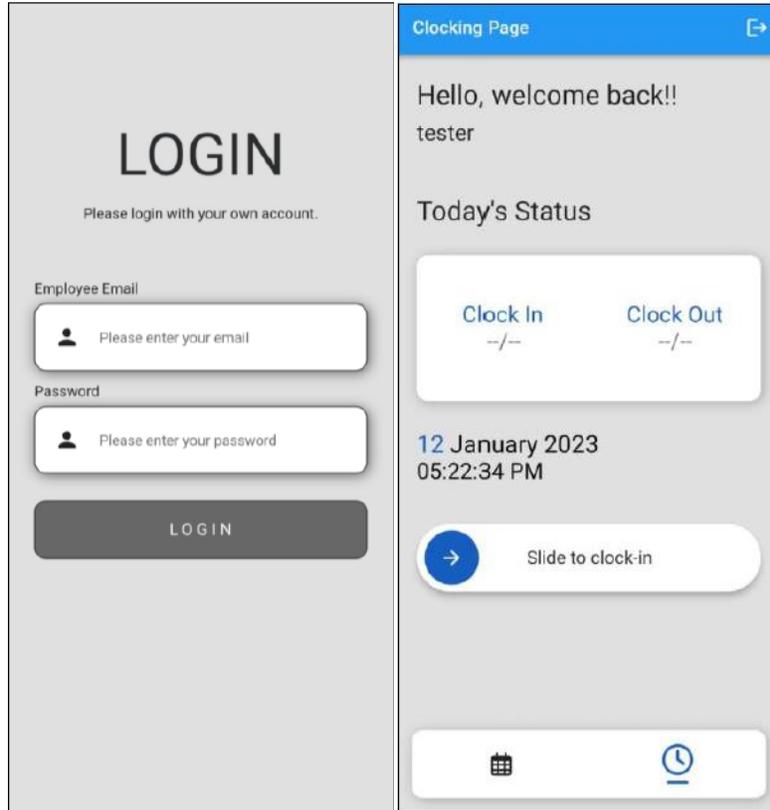


Figure 3: Interface for Login Page and Clocking Page

In the meantime, Figure 4 illustrates instances of the Clock Log History section, which facilitates the user in accessing their past attendance records. Users do not possess the capability to modify or alter their recorded clock-in or clock-out data. The default display of records is organised by month, allowing users to choose their desired month and year through the "Month Picker" option.



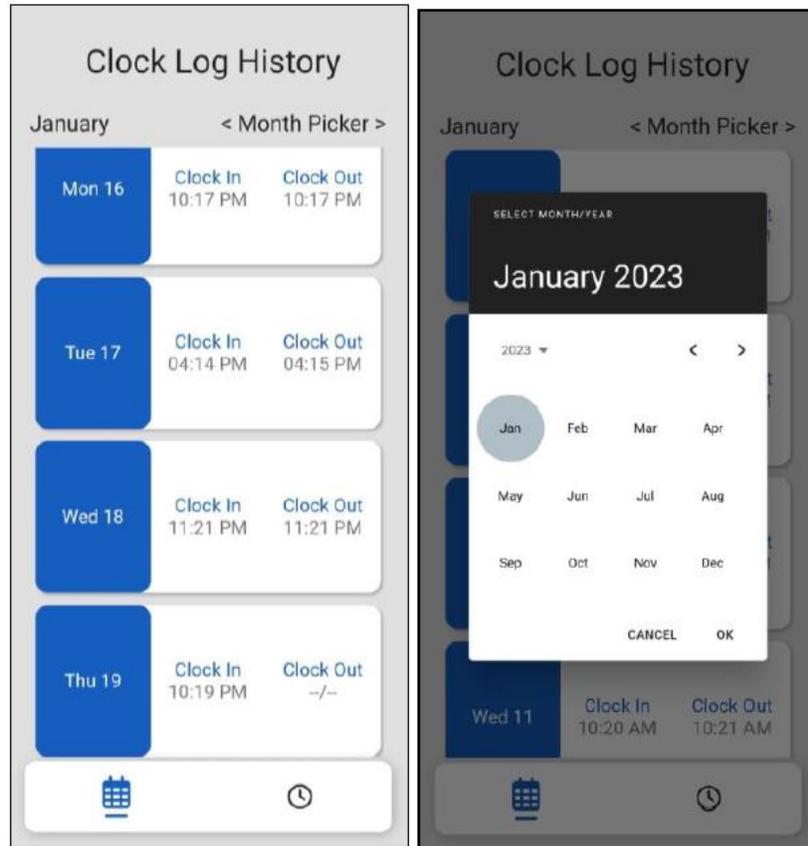


Figure 4: Interface for the Clock Log History section and Month Picker feature.

FINDINGS AND DISCUSSIONS

Functionality Test

After the completion of the application's development, a series of testing processes were carried out. Both functionality and usability testing are conducted in order to assess the application. Functionality is a bridge between developers, analysts, and testers that focuses on comprehending system requirements. In order to identify test scenarios for the application, it is imperative for the developer to possess a comprehensive comprehension of the functional requirements of the programme. A test case typically comprises several key components, including a comprehensive description of the test scenario, the anticipated outcomes or expected results, the actual results obtained during the test execution, and an indication of the test's status, denoting whether it passed or failed.

Table 1 below shows the functionality testing results for the developed application.



Table 1 Functionality testing results.

Test Case	Description	Expected Results	Actual Results	Status
Login Functionality	Check the response by entering a valid email and password.	The login needs to be successful.	Successfully log in with email and password.	Pass
	Check the response on entering an invalid email and password.	A message should appear as an alert.	A message appears for an invalid email and password.	Pass
	Check the response on the empty email and password field (validation).	A message should appear as an alert.	A message appears for an empty field email and password.	Pass
Navigation Bar / Sections	Check on changing the section between the Clocking Page and the Clock History Log.	Able to navigate between both sections.	Successfully navigate between both sections.	Pass
Clocking Page Section	Check on the clock-in/out slider.	Able to slide.	Successfully slide.	Pass
	Check on the clock-in/out time being displayed.	Should display the employee today's status.	Employee today's status displayed.	Pass
	Check on the location being displayed.	Should display the location.	Location was displayed.	Pass
	Check on the alert message after clock-out.	A message should appear after clock-out.	A message appears after clock-out.	Pass
	Check on the changes from clock-in to clock-out on the slider.	Should change after clock-in.	Successfully changed after lock-in.	Pass
Clock Log History Section	Check on the Month's Picker button.	Able to click Month's Picker button.	The button was clickable.	Pass
	Check on the ability to change the month.	Able to change the month.	The month was changed.	Pass
	Check on the ability to change the year.	Able to change the year.	The year was changed.	Pass



	Check on the Cancel button	Able to click the cancel button.	The cancel button was clickable.	Pass
Logout Functionality	Check on the ability to logout	The logout must be successful	Successfully logout from the application.	Pass

Usability Test

Usability testing is a form of software testing that centres on the assessment of a product's user-friendliness and the holistic user experience. The application underwent evaluation through the utilisation of usability testing on a sample size of 32 participants. The objective of usability testing is to assess if the attendance system meets the established usability standards of end-users. In order to conduct testing, participants are provided with the actual application. Participants are afterwards requested to fill out the questionnaire provided through the Google form. The researcher meticulously documented and analysed every response to the questionnaire, and Table 2 shows the comprehensive presentation of all pertinent information derived from the findings. The questionnaire consists of four questions in which respondents are required to score their experience using a five-point Likert scale. The scale includes the following options: The response options range from Strongly Disagree (1) to Strongly Agree (5).

Table 2 Question from Usability Testing.

Question	1	2	3	4	5	Mean
1. The system is easy to use.	0%	0%	3.1%	37.5%	59.4%	4.5625
2. The system is understandable.	0%	0%	0%	40.6%	59.4%	4.5938
3. The system is simple and neat.	0%	0%	0%	37.5%	62.5%	4.625
4. The system is user friendly.	0%	0%	3.1%	31.3%	65.6%	4.625

The data shown in Table 2 indicates that there is a favourable response towards the features of the application. The application was developed using a user-centric approach and with the aim of reducing the complexity of the learning process. The application's functionality is enhanced by the real-time records made possible through the seamless integration with Firebase. Regarding questions 1 and 4, the neutral outcomes can be attributed to the respondents' lack of familiarity with the system. This study marks the initial use of a mobile application-based attendance record system among the respondents. Overall, it can be concluded that the application meets the usability criteria based on mean value obtained from the results.



CONCLUSION AND RECOMMENDATIONS

In summary, an employee attendance application can serve as a valuable instrument for monitoring employee time and attendance; nevertheless, it is crucial to select a solution that aligns with the unique requirements of the organisation. There exist numerous methods for improving the employee attendance system. Nevertheless, when organizations are in the process of choosing an application, they must take into account various criteria including accessibility, clock-in options, reporting capabilities, interaction with other systems, offline support, security, and cost. Hence, there is optimism that the developed application will offer assistance to small or medium-sized firms or organisations in the monitoring of staff attendance, as an alternative to conventional methods.

Finally, in order to cater to diverse employee preferences, it is suggested that the application be enhanced to incorporate a wider selection of clock-in options, including facial recognition or biometric data. In addition, to streamline the management of employee data and reduce administrative burdens, it is recommended to interface the application with additional human resources software or payroll systems. Developing an offline functionality for the application will enable users to perform clock-in and clock-out operations even in situations when the Internet connectivity is limited or unavailable.

FUNDING

There was no specific grant for this research from any governmental, commercial, or not-for-profit funding agencies.

ACKNOWLEDGMENTS

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

CONFLICT OF INTEREST DISCLOSURE

The authors declared that they have no conflicts of interest to disclose.

REFERENCES

- Ahmed, F. Y. H., Aik, K. L. T., Radzi, A. S., & Salleh, M. D. (2019). Develop Attendance Management System with Feedback and Complaint Management Function. *2019 IEEE 7th Conference on Systems, Process and Control (ICSPC)*, 248–252. <https://doi.org/10.1109/ICSPC47137.2019.9068041>
- Dart overview. (n.d.). Retrieved August 18, 2023, from <https://dart.dev/overview>
- Iio, J. (2016). Attendance Management System Using a Mobile Device and a Web Application. *NBiS 2016 - 19th International Conference on Network-Based Information Systems, September 2016*, 510–515. <https://doi.org/10.1109/NBiS.2016.44>



Kaur, A., Bhatia, M., & Stea, G. (2022). A Survey of Smart Classroom Literature. *Education Sciences*, 12(2), 1–30. <https://doi.org/10.3390/educsci12020086>

Kaur, G. (2018). *Attendance Tracking by Facial Recognition*. <https://scholarworks.gvsu.edu/cgi/viewcontent.cgi?article=1314&context=cistechlib>

Muhtahir O., O., Adeyinka O., A., & Kayode S., A. (2013). Fingerprint Biometric Authentication for Enhancing Staff Attendance System. *International Journal of Applied Information Systems*, 5(3), 19–24. <https://doi.org/10.5120/ijais12-450867>

Oo, S. B., Oo, N. H. M., Chainan, S., Thongniam, A., & Chongdarakul, W. (2018). Cloud-based web application with NFC for employee attendance management system. *2018 International Conference on Digital Arts, Media and Technology (ICDAMT)*, 162–167. <https://doi.org/10.1109/ICDAMT.2018.8376516>

