MOBILE APPLICATION DEVELOPMENT (Android Only) (Go-Manage - A One Step Solution for Managing Students' Data)

Harjeet Singh, Department of Mathematics, R.D Engineering College, Duhai, Ghaziabad, U.P., India 201206
Email- harjeet.singh@gmail.com

ABSTRACT

The student management app is a comprehensive platform designed to streamline and improve the management of student information and academic processes. The app is built using Flutter and Firebase as the backend, which provides a robust and secure solution for managing student data. The app includes features such as user authentication, student profiles, course management, attendance tracking, grading and assessment, and analytics and reporting. Additionally, the app also includes a leave application feature, which allows students to submit leave requests and instructors to approve or reject them. The end output of the app is a centralized platform for students, instructors, and administrators to interact and communicate, which enhances the overall educational experience and reduces administrative workload. The student management app provides real-time access to important information, improved communication, and increased student engagement, making it an indispensable tool for colleges.

In recent years, the need for efficient and effective student management systems has become increasingly important in educational institutions. With the rise of digital technology, a student management application built using Flutter offers an excellent solution for educators and administrators looking to manage student information, grades, and schedules. This application provides a centralized platform for recording and tracking student performance, attendance, and grades, while enabling communication between teachers and parents. Built using the Flutter framework, the application is responsive, interactive, and capable of running seamlessly on both Android and iOS platforms. This report presents an overview of the features, functionality, and benefits of a student management application built using Flutter, including student information management, attendance management, assignment management, class scheduling, grade management, parent communication, and reporting. The report concludes that a student management application built using Flutter offers an effective and efficient solution for managing student information, improving academic performance, and enhancing communication between teachers, parents, and students.

KEY WORD:- Cross-Platform Mobile application development, IDE, Android development, iOS development, Flutter, Dart.

INTRODUCTION

1.1 Mobile applications are having a progressively more significant role in our day to day lives. Ever since November 2016, there has been more network traffic made by mobile devices (48.19%) compared to desktops or laptops (47%). To dispense it to most of the users, a mobile application needs to familiarize itself with two independent platforms which are Android and iOS. These two platforms share immense dissimilarities which often necessitate different skill sets for developing. For example Java or Kotlin for Android and Object-C or Swift for iOS. Hence, developers and companies usually struggle to deal with the complex nature involved in developing cross-platform applications. On March 15, Facebook introduced an open-source cross-platform JavaScript framework called React Native which strives to solve the aforesaid problem. In addition to its effective developing process, developers use React framework extensively because of its non-complex and easy nature. Further, Google announced another mobile SDK named Flutter in the latter half of 2016. Inspired by React Native, Flutter applications can also run equally on both platforms, consequently decreasing the cost and complexity of application creation across iOS and Android. Flutter is fully built from scratch and at the time of scripting this study (Aug 2017), only Google uses it for commercial projects. Cross-platform frameworks

that show resemblance to React Native and Flutter, are discussed and implemented by various companies numerous times formerly. Still, neither of them suffices to satisfy the requirement of industrial development. In spite of the ineffective precursors, React Native and Flutter, which are backed up by Facebook and Google, draw attention and people are optimistic about its prospects.

1.2 Flutter is a cross-platform framework that targets developing high-performance mobile applications. Flutter was publicly released in 2016 by Google. Besides running on Android and iOS flutter applications also run on Fuschia. Flutter is chosen as Google's application-level framework for its next-generation operating system. Flutter is exceptional because it is dependent on the device's OEM widgets rather than consuming web views. Flutter uses a high-performance rendering engine to render each view component using its own. This provides a chance to build applications that are as highperformance as native applications can be. In view of architecture, the engine's C or C++ code involves compilation with Android's NDK and LLVM for iOS respectively, and during the compilation process, the Dart code is compiled into native code. Hot reload feature in Flutter is called Stateful hot reload and it is a major factor for boosting the development cycle. Flutter supports it during development. Stateful hot reload is implemented by sending the updated source code into the running Dart Virtual Machine (Dart VM) without changing the inner structure of the application, therefore the transitions and actions of the application will be well-preserved after hot reloading.

1.3 In Flutter, every application is written with the help of Dart. Google has developed and maintained a programming language called Dart. It is extensively used inside Google and it has been verified to have the proficiency to develop enormous web applications, such as AdWords. Originally Dart was developed to replace and succeed JavaScript. Thus, it implements most of the important characteristics of JavaScript's next standard (ES7), such as the keywords "async" and "await". Nonetheless, to attract developers that are not acquainted with JavaScript, Dart has a Java-like syntax. Flutter application renews the view tree on every new frame even when few other systems use reactive views. This behavior leads to a drawback that many objects, which might survive for a singular frame, will be created. As Dart is a modern programming language, it is optimized to handle this scenario in memory level with the help of "Generational Garbage Collection".

LITERATURE REVIEW

A student management app for colleges is a cutting-edge solution for streamlining various processes related to student administration and academic management. This app aims to provide a unified platform for students, teachers, and administrators to access and manage important information related to coursework, schedules, grades, and more. With this app, colleges can automate various manual processes, reduce administrative workload. The app is designed to be user-friendly and accessible, offering students and staff the ability to access and manage information on the go, from any device. The student management app is the perfect solution for colleges looking to enhance their operations and improve the overall educational experience for their students.

Additionally, the student management app also provides an efficient platform for students to interact with their peers, instructors, and administrators. They can participate in online discussions, collaborate on projects, and access important course material, all from one central location. The app also integrates with other systems, such as student financials and human resources, to provide a comprehensive view of each student's progress and involvement at the college. Furthermore, the app is highly secure and ensures that sensitive information is protected and only accessible to authorized users.

The student management app also helps colleges to make data-driven decisions by providing real-time analytics and reporting capabilities. College administrators can

monitor student progress, track attendance, and analyze trends in student performance. This information can then be used to make informed decisions and improve the overall quality of education offered at the college.

In conclusion, a student management app for colleges is a powerful tool that can help colleges to manage and streamline their operations, while also providing students and staff with a convenient and accessible platform for accessing information and interacting with each other. With the right student management app, colleges can improve the overall educational experience for their students and achieve their goals of providing quality education.

PROPOSED SOLUTION

A proposed solution for a student management app built using Flutter and Firebase as the backend could include the following features:

● User Authentication: The app should have secure authentication and authorization capabilities to ensure that only authorized users can access the app. This can be achieved using Firebase Authentication, which integrates with Flutter easily.

● Student Profiles: Each student should have a personal profile that includes their academic information, such as coursework, grades, and schedules. This information can be stored in Firebase Firestore, which is a NoSQL database that integrates with Flutter seamlessly.

● Course Management: The app should allow students to view their course schedules, access course materials, and participate in online discussions with their peers and instructors. This information can be stored in Firebase Firestore, and retrieved in real-time using Flutter's StreamBuilder.

● Attendance Tracking: The app should allow instructors to take attendance and monitor student attendance in real-time. This information can be stored in Firebase Firestore and retrieved using Flutter's StreamBuilder.

● Grading and Assessment: The app should provide an efficient platform for instructors to grade assignments and provide feedback to students. This information can be stored in Firebase Firestore, and retrieved in real-time using Flutter's StreamBuilder.

● Leave Request: Students should be able to submit leave requests from within the app, which will be sent to their instructors for approval. This information can be stored in Firebase Firestore, and retrieved in real-time using Flutter's StreamBuilder.

○ The app should provide real-time notifications to students and instructors regarding the status of leave requests. This can be achieved using Firebase Cloud Messaging, which integrates with Flutter easily.

○ Students should be able to view a history of their leave requests, including the dates and reasons for each request, as well as the status of each request.

● Analytics and Reporting: The app should provide real-time analytics and reporting capabilities to allow college administrators to monitor student progress and make datadriven decisions. This information can be retrieved from Firebase Firestore using Flutter's StreamBuilder and displayed in an intuitive and easy-to-use interface.

In conclusion, a student management app built using Flutter and Firebase as the backend can provide a comprehensive and efficient solution for managing student information and academic processes. By leveraging the power of Flutter and Firebase, the app can provide real-time access to important information and enable students, instructors, and administrators to interact and collaborate effectively.

OUTPUT

The end output of a student management app is a comprehensive and efficient platform for managing student information and academic processes. The following are some of the key benefits that can be expected from a well-designed student management app:

● Improved Communication: The app provides a centralized platform for students, instructors, and administrators to interact and communicate with each other, which can greatly improve the overall educational experience.

● Streamlined Processes: By automating various manual processes, such as leave requests and course management, the app can reduce administrative workload and free up time for instructors and administrators to focus on more important tasks.

● Real-time Access to Information: The app provides real-time access to important information, such as grades, schedules, and course materials, which can greatly enhance the student experience.

● Analytics and Reporting: The app provides real-time analytics and reporting capabilities, which allow college administrators to make data-driven decisions and monitor student progress.

● Increased Student Engagement: The app provides a platform for students to interact with their peers, instructors, and administrators, which can increase student engagement and help to create a more collaborative and supportive educational environment.

In conclusion, a well-designed student management app has the potential to significantly improve the educational experience for students, while also streamlining processes and reducing administrative workload for instructors and administrators. The end output of such an app is a comprehensive and efficient platform for managing student information and academic processes, which can greatly enhance the overall quality of education offered at a college.

APPLICATION ILLUSTRATION

To further illustrate the capabilities and features of the student management app, we have included a series of screenshots that showcase the app's user interface and key functionalities. These screenshots provide a visual representation of the app's design and layout, and demonstrate how the app can be used to streamline and simplify student management processes.

The screenshots highlight some of the app's key features, such as the student dashboard, which provides quick and easy access to important information and resources, the leave application module, which allows students to submit leave requests and track the status of their requests, and the analytics and reporting functionality, which provides valuable insights into student performance and academic progress.

Additionally, the screenshots showcase the app's user-friendly design, which makes it easy for students, instructors, and administrators to navigate the app and find the information they need. The app's intuitive and responsive interface ensures that users can access the information and features they need quickly and easily, without any unnecessary delays or distractions.

In conclusion, the screenshots provide a visual representation of the student management app's capabilities and features, and demonstrate how the app can be used to improve the quality of education and streamline student management processes. The screenshots provide a glimpse into the app's potential and showcase the many benefits it can offer to colleges and universities around the world.

The screenshots also demonstrate the app's scalability and versatility, showcasing how it can be customized to meet the unique needs and requirements of different colleges and universities. This flexibility ensures that the app can be tailored to the specific needs of each institution, providing a truly customized solution.

Here are some app screenshot(s)[1][2] and their detailed description in chronological order.

Screenshot[1].

The login screen of a student management app is an essential element of the app's user interface. It serves as the entry point for users, allowing them to authenticate their identity and access the features and functions of the app. The login screen

typically consists of a form that prompts users to enter their login credentials, such as a username and password.

Depending on the app's design and security requirements, users may also be asked to provide additional information, such as a security question or a two-factor authentication code.

To ensure a smooth and efficient login process, the login screen should be designed with the user in mind. It should be easy to navigate and use, with clear instructions and visual cues to guide users through the login process. The form should also be optimized for different screen sizes and devices, so that users can access the app from anywhere, on any device.

Security is also a crucial consideration for the login screen of a student management app. The app should implement best practices for data security, such as encryption, password policies, and secure session management. This will help to protect users' sensitive information and prevent unauthorized access to the app.

Overall, the login screen of a student management app is a critical component of the user experience. By prioritizing usability and security, app developers can create a login screen that helps users easily and securely access the features and functions of the app

Screenshot[2].

The token ID generation screen for user registration is an essential component of many modern web applications. It is a part of the user registration process, and its primary purpose is to generate a unique token ID that is associated with each user account. This token ID can then be used to verify the user's identity and authenticate their login in subsequent sessions.

The token ID generation screen typically appears during the user registration process, after the user has entered their basic account information such as their name and email address. Once the user submits their registration details, the token ID generation screen will generate a unique identifier for the user account, which is usually a randomly generated alphanumeric code.

The generation of the token ID should be performed using a secure and random algorithm to ensure that the tokens are difficult to guess or replicate. The token ID should be long enough to provide sufficient entropy, but not so long as to be impractical to use. In general, a token ID between 16 and 64 characters in length is sufficient for most applications.

Once the token ID has been generated, it should be stored securely and associated with the user's account. This can be accomplished by storing the token ID in a secure database, along with the user's account details, and using it to verify the user's identity during subsequent login attempts.

Overall, the token ID generation screen for user registration is a critical component of modern web applications. By providing a secure and reliable mechanism for generating unique identifier codes for user accounts, it helps to ensure that the user's identity is protected and that their account is kept secure

CONCLUSION

In conclusion, the student management app is a powerful solution that addresses the need for an efficient and comprehensive platform for managing student information and academic processes. The app's integration with Flutter and Firebase provides a secure and scalable platform that is well-suited to meet the needs of colleges and universities. The app's features, such as user authentication, student profiles, course management, attendance tracking, grading and assessment, and analytics and reporting, provide a comprehensive set of tools for managing student data. Additionally, the app's leave application feature helps to streamline the process of submitting and approving leave requests, further improving the overall educational experience.

In today's fast-paced educational environment, the student management app provides a critical solution that helps colleges to stay ahead of the curve. By automating various

manual processes and providing real-time access to important information, the app helps to streamline processes, reduce administrative workload, and enhance the overall quality of education offered at a college. The student management app is a must-have tool for any college that is looking to improve its student management processes and provide a better experience for students, instructors, and administrators.

In addition, the student management app can also improve student engagement and foster a more collaborative educational environment. With real-time access to information, students are better able to stay informed and engaged, which can improve their overall academic performance. The app's integration with Firebase Cloud Messaging provides real-time notifications and updates, keeping students informed and in-the-loop at all times. Moreover, the app's analytics and reporting capabilities provide valuable insights into student progress and academic performance, allowing colleges to make data-driven decisions and monitor student success. This information can be used to identify areas of improvement, develop new strategies for student engagement, and evaluate the effectiveness of existing programs and initiatives. In conclusion, the student management app is a comprehensive solution that provides many benefits to colleges and universities, including improved communication, streamlined processes, real-time access to information, increased student engagement, and valuable analytics and reporting capabilities. The app is a must-have tool for any college that is looking to improve its student management processes and provide a better experience for students, instructors, and administrators.

FUTURE WORK

The student management app is a powerful platform that has the potential to be further improved and expanded in the future. Some of the key areas for future work include:

Integration with other educational systems: The student management app can be integrated with other systems, such as learning management systems, student information systems, and payment systems, to provide a more comprehensive solution. This integration will enable the app to offer a more streamlined and efficient experience for students, instructors, and administrators.

● Mobile App Development: The student management app can be further improved by developing a mobile app that provides students, instructors, and administrators with real-time access to information and features from their mobile devices. This will make the app more accessible and convenient to use, further improving the overall educational experience.

● Personalized Dashboards: The app can be further customized to provide each user with a personalized dashboard that provides them with quick access to the information and features that they need. This customization will enhance the user experience and make it easier for users to find the information they need quickly and easily.

● Predictive Analytics: Predictive analytics can be used to provide valuable insights into student performance and predict future performance. This information can be used to provide students with personalized feedback and guidance, as well as identify areas of improvement and develop new strategies for student success.

● Improved Security: As technology continues to evolve, it is important to ensure that the student management app remains secure and protected against data breaches and other cyber threats. Future work in this area will be focused on strengthening security protocols and ensuring that sensitive student data remains protected at all times.

● Multi-Language Support: The student management app can be further improved by adding support for multiple languages. This will make the app more accessible to a wider range of users and help to promote diversity and inclusivity within the educational community.

● Real-time Collaboration: The student management app can be enhanced by adding real-time collaboration features that enable students, instructors, and administrators to work together in real-time, regardless of their location. This will foster a more collaborative and inclusive educational environment and help to improve communication and collaboration between all stakeholders.

In conclusion, the student management app has the potential to be further improved and expanded in the future. Future work will focus on enhancing the app's capabilities and improving the overall educational experience for students, instructors, and administrators. The future looks bright for the student management app, and its continued evolution will be critical to ensuring its long-term success and impact.

REFERENCES

● React Native vs Flutter, Cross-Platform Mobile Application Framework, Thesis March 2018- Wenhao Wu.
● A. Smith. (2019). Why businesses should start focusing on google's flutter and fuchsia, [Online]. Available: https://medium.com/swlh/whybusinesses - should - start - focusing - on - googles - flutter - and - fuchsia-48e16820f2a9 (visited on 01/21/2019).
● M. Palmieri, I. Singh, and A. Cicchetti, "Comparison of cross-platform mobile development tools," 2012 16th International Conference on Intelligence in Next Generation Networks, Researchgate.net, 2012, p. 8.
● S. Xanthopoulos and S. Xinogalos, "A comparative analysis of cross platform development approaches for mobile applications," University of Macedonia, academia.edu, 2013, p. 7. [Online]. Available: https://dl. acm.org/doi/abs/10.1145/2490257.2490292.
● Google. (2019). Google flutter, [Online]. Available: https://flutter. dev/ (visited on 12/03/2019).
● B. D. Coninck. (2019). Flutter versus other development frameworks: A ui and performance experiment, [Online]. Available: https : / / blog . codemagic . io / flutter - vs - ios - android - react native - xamarin/ (visited on 02/06/2020).
● M. Bellinaso. (2018). Flutter: The good, the bad and the ugly, [Online]. Available: https://medium.com/asos-techblog/flutter-vs-reactnative-for-ios-android-app-developm ent-c41b4e038db9 (visited on 02/06/2020).
● A clean approach to Flutter Development through the Flutter Clean architecture package, IEEE 2019, Shady Boukhary, Eduardo Colemenares.
● Exploring end user's perception of Flutter mobile apps, Malmo University Nov 2019- Dahl, Ola.
● Flutter for Cross-Platform App and SDK Development, Metropolia University Thesis May 2019- Lucas Dagne.
● Cross-Platform Framework comparison- Flutter vs React Native.
● Flutter Native Performance and Expressive UX/UI, paper 2019- Tran Thanh.
● Olsson, M.: "A Comparison of Performance and Looks between Flutter and Native Applications". , Karlskrona Sweden (2020).
● S. Amatya, "Cross-platform mobile development: An alternative to native mobile development," Linn´euniversitetet, Fakulteten f˙or teknik (FTK), Institutionen for datavetenskap (DV), diva2:664680: Diva, 2013, p. 67. 40

● H.Hussain, N.Faseeh, Q.Ali, I.Farah, "Comparative Study of Database Tools for Android Application: A Bird's Eye View," in Proc. of the 12th International Conference on Internet (ICONI 2020), 2020.
● I. Dalmasso, S. K. Datta, C. Bonnet, and N. Nikaein, Survey, comparison and evaluation of cross-platform-mobile-application-development-tools, 2013.
● Google. (2019). Google flutter sdk releases, [Online]. Available: https:

● //flutter.dev/docs/development/tools/sdk/releases (visited on 01/21/2019).
● G. Flutter. (2020). Flutter setup, [Online]. Available: https://flutter. dev/docs/getstarted/install (visited on 04/30/2020).
● A. Inc. (2020). Xcode setup, [Online]. Available: https://developer. apple.com/xcode/ (visited on 04/30/2020). W. Danielsson, "React native application development," Diva, diva2:998793: Faculty of Computer science LIU, 2016, p. 70.

- B. Armour. (2018). 5 key benefits of native mobile app development, [Online]. Available: https://clearbridgemobile.com/benefits- of native-mobile-app-development/ (visited on 05/03/2020).
- C. Griffith. (2019). What is cross-platform app development, [Online]. Available: https://ionicframework.com/resources/articles/whatis-cross-platform-app-development (visited on 02/13/2020).
- C. Software. (2019). What is flutter here is everything you need to know, [Online]. Available: https : / / medium . com / @concisesoftware / what - is- flutter- here- is-everything- you- should- know- faed3836253f (visited on 05/07/2020).
- G. Flutter. (2020). Faq flutter, [Online]. Available: https://flutter.
- dev/docs/resources/faq (visited on 02/13/2020).
- Google. (2020). Dart, [Online]. Available: https://dart.dev/ (visited on 06/05/2020).
- M. Satei, "Ott video-oriented mobile applications development using cross platform ui frameworks," KTH Royal Institute of Technology, School of electrical engineering and computer science, diva2:1343759: Diva, 2018 , p. 90. C. Software. (2019). What is flutter? Here is everything you should know, [Online]. Available: https : // medium . com / @concisesoftware / what - is- flutter- here- is- everything- you- should- know-faed3836253f (visited on 03/05/2020).
- Google. (2020). Material io components, [Online]. Available: https : / / material.io/components (visited on 05/03/2020).
- Stackoverflow. (2019). Stackoverflow survey 2019, [Online]. Available: https: //insights.stackoverflow.com/survey/2019 (visited on 06/01/2020).
- ——, (2020). Stackoverflow survey 2020, [Online]. Available: https:// insights.stackoverflow.com/survey/2020 (visited on 06/01/2020).
- G. Flutter. (2020). Google flutter animations, [Online]. Available: https:
- //flutter.dev/docs/development/ui/animations (visited on 03/12/2020).
- **Dharamveer**, Samsher, Singh D.B., Singh A.K., Kumar N. (2019) "Solar Distiller Unit Loaded with Nanofluid—A Short Review". In: Kumar M., Pandey R., Kumar V. (eds) Advances in Interdisciplinary Engineering. Lecture Notes in Mechanical Engineering. Springer, Singapore. pp 241-247, Paper Published. **Scopus Index,** Springer Publication. https://doi.org/10.1007/978-981-13-6577-5_24
- **Dharamveer,** Samsher "Comparative Analysis of Energy Matrices and Enviro-economics for Active and Passive Solar Still". Journal Materials Today proceedings, Elsevier publication. https://doi.org/10.1016/j.matpr.2020.10.001
- **Dharamveer,** Samsher, Anil Kumar "Performance analysis of N-identical PVT-CPC collectors an active single slope solar distiller with a helically coiled heat exchanger using CuO nanoparticles", Water supply, Vol. 22 (201) 02 1306-1326, October 2021 https://doi.org/10.2166/ws.2021.348
- **Dharamveer,** Samsher, Anil Kumar "Analytical study of photovoltaic thermal (PVT) compound parabolic concentrator (CPC) active double slope solar distiller with helical coiled heat exchanger using CuO Nanoparticles" Desalination and water treatment, vol. 233 (2021) 30–51 https://doi.org/10.5004/dwt.2021.27526