Attendance Management Module for An Academic Department Using React Native Framework

Swati Mishra[1], Neha Mehra[2]
[1]Shri Govindram Seksaria Institute of Technology and Science, Indore, India, mishraswati621@gmail.com
[2]Shri Govindram Seksaria Institute of Technology and Science, Indore, India, mehra.neha40@gmail.com

ABSTRACT

Attendance management is an aspect of academic institutions that directly impacts student engagement and performance. The proposed solution aims to streamline the attendance tracking process, enhance efficiency, and improve student-teacher interactions. By leveraging the cross-platform capabilities of React Native and the comprehensive features of Moodle, the attendance management module offers a user-friendly interface accessible through mobile devices. The system incorporates real-time attendance tracking, automated record-keeping. The research paper details the design considerations, technical implementation, and discusses the potential benefits and challenges of deploying such a solution. This research paper presents the design and implementation of an attendance management module for an academic department using the React Native framework with integrated Moodle.

Key words: Attendance management, React Native, Moodle, mobile application, academic department

1. INTRODUCTION

Smartphones have emerged as the most significant consumer invention of the 21st century, revolutionizing various aspects of our lives. Using the power of mobile technology, attendance tracking systems can provide real-time updates to students, [11] teachers, and administrators, leading to improved student engagement, accountability, and convenience in managing attendance records. To address these challenges, this research paper proposes a mobile attendance system based on model the Android platform. [2] The system offers a user-friendly inter- face for both students and staff members, ensuring easy access and efficient management of attendance records. Furthermore, the system has been designed to seamlessly integrate with the Department Moodle, giving a smooth and well-organized experience [17].

1.1 Background

W Traditional methods of managing attendance records in academic institutions present significant difficulties and limitations. Teachers have to manually record attendance for each student, increasing the likelihood of mistakes and inaccuracies. The dependencies on manual procedures, such as paper-based attendance sheets lead to inefficiencies, errors. Additionally, accessing attendance records becomes a complex task as physical storage or limited sharing options hinder effective communication among teachers, students, and administrators. This lack of accessibility restricts the ability to identify attendance patterns or promptly address attendance-related issues. These outdated approaches hinder the overall student as well as teachers experience and create additional administrative burdens.

Traditionally students don't have access directly to track their attendance or staying updated with attendance report. Furthermore, the administrative burdens associated with manual attendance management are substantial. The process of collecting and generating attendance reports consumes valuable administrative time and resources diverting attention from other essential tasks. There is a clear need for a modern and effective system to manage attendance, one that addresses these challenges and enhances the efficiency of the process. To overcome these challenges, the development of an attendance management app using modern technologies becomes crucial [3].

Such an app offers a seamless and convenient solution, allowing teachers to digitally record attendance, reducing errors and streamlining the process. Real-time accessibility of attendance data empowers teachers, students, and administrators to track records, generate reports, and make data-driven decisions promptly.

1.2 Objectives

The main objective of this paper is to design and implement an Attendance Management module that addresses several key goals. the aim of the work is to provide a convenient and accessible way for Administrator, teachers and students to access attendance records. Focus is to create a user-friendly system that is intuitive for both teachers and students to

navigate and utilize effectively. The system will enable teachers to filter the attendance sheet based on specific criteria and download it in PDF or Excel format, facilitating streamlined record-keeping and data analysis. Additionally, the module will incorporate a feature to categorize students based on their attendance, allowing for easy identification of students with attendance below a specified threshold, such as 75%. These objectives collectively contribute to improving the overall attendance management process, fostering better student engagement, and enabling more informed decision-making by teachers and administrators. By leveraging modern technologies, this paper seeks to enhance the traditional process of attendance tracking, making it more efficient and accurate [16].

1.3 Scope and Limitations

A. *Scope*

This paper focuses on the design and implementation of an attendance management module for academic departments. The main topics covered include integrating the module with the Moodle learning management system and building it using the React Native framework. The study involves developing a cross-platform mobile application for attendance management, involving tasks such as data management, user interface development, and synchronization with the Moodle platform. [4]

B. *limitations:*

Tracking Attendance and Record-Keeping: The study paper examines the features connected to attendance tracking, in-depth reporting, real-time monitoring, and auto- mated record-keeping. It seeks to make attendance management more efficient while also providing accurate and current attendance records.

Mobile-Only Functionality: The focus on React Native may restrict the module's accessibility to mobile devices.

Limited Integration: The integration of the attendance management module with Moodle, a popular learning management system, may limit its usage to institutions or departments that already use Moodle. Institutions that utilize different learning management systems may not be able to implement or benefit from this specific integration.

Data Synchronization Challenges: Ensuring real-time synchronization of attendance data between the mobile application and the Moodle system may pose technical challenges. Maintaining consistency and reliability in data updates across platforms could be a limitation.

Network Connectivity Dependency: Mobile applications heavily rely on network connectivity. In scenarios where network connectivity is poor or unavailable, the attendance management module's functionality may be impacted, leading to potential disruptions in attendance tracking.

Security and Privacy Concerns: As attendance records may contain sensitive data, the design and implementation of the module need to address security and privacy concerns adequately. Failure to do so could result in unauthorized access to attendance data or potential privacy breaches.

Scalability: The scalability of the attendance management module, particularly in large academic departments or institutions with a significant number of users, could be a limitation. The module should be designed to handle a substantial load of concurrent users and efficiently manage attendance records without compromising performance.

Customization and Flexibility: The attendance management module might lack the flexibility to accommodate diverse attendance policies and requirements of different academic departments or institutions. Customization options should be carefully considered to address these variations.

Maintenance and Support: The long-term maintenance and support of the attendance management module, including updates, bug fixes, and user support, could be a challenge. Adequate resources and a dedicated support team may be required to ensure the module's continued functionality and address user inquiries or issues.

2. SYSTEM TECHNOLOGY

2.1 React Native Technology

React Native (RN) is an open-source front-end framework based on JavaScript. It combines the advantages of web and native front-end application development, allowing the creation of native applications for both Android and iOS systems [20]. Within the framework, React is utilized to manipulate native UI components, replacing the traditional rendering of DOM elements [12]. RN employs JavaScript as its development [13] language [10], offering a flexible application development process through component-based lifecycle mechanisms and attribute-driven data interaction. Despite being developed in JavaScript, RN compiles into a native application, ensuring optimal performance. With RN, the concept of 'one set of code, running in multiple places' is realized, simplifying collaboration between teams as developers only require a basic understanding of JavaScript [6].

2.2 Moodle

Moodle (Modular Object-Oriented Dynamic Learning Environment) is basically a learning platform that has been created with the purpose of offering educators, administrators, and learners a unified, dependable, and comprehensive system to establish customized learning environments. It aims to provide a secure and integrated solution for all individuals involved in the process of education. [5] [9]

3. LITERATURE REVIEW

Comparative analysis of Angular.js and React.js would involve synthesizing existing research and articles that compare these two popular JavaScript frameworks. The review would aim to provide an in-depth understanding of the similarities, focusing on key factors such as performance, ease of use, scalability, community support, and versatility. the advantages and disadvantages of each framework and provide insights into their practical applications in web development. The goal of the review would be to summarize and evaluate current knowledge on the topic and inform future research and development in this area.[1]

The work presented by S. Z. Faidah, M. B. Ulum, and P. Handayani explored the use of smartphones and mobile applications to replace manual attendance systems in academic institutions [3]. They proposed an automated attendance recording system using a mobile Android application, simplifying the process for lecturers and providing convenient attendance management. The system allowed attendance records to be easily retrieved, previewed, and printed, with reports sent to students via email, SMS, or in Excel and PDF formats. This research showcased the potential of technology in streamlining attendance tracking in educational settings.
An excellent style manual and source of information for science writers is [9].

The author proposed that the demand for faster and improved applications and tools has significantly increased in recent times. JavaScript frameworks have emerged as valuable assets due to their popularity, ease of use, and seamless integration, leveraging new technologies to benefit both developers and users. Choosing the most suitable framework for a paper is ultimately the decision of the manager. However, this analysis is focused on comparing two major JavaScript frameworks: Vue.js and React.js. [14]

Would examine the main proposed the differences and similarities between these two JavaScript frameworks. The study would evaluate the performance, ease of use, scalability, community support, and versatility of both frameworks in the context of web development applications. The summary would aim to provide a concise overview of the findings of the research and draw conclusions about the relative strengths and weaknesses of Angular and Vue.js. The paper also suggest which framework might be more suitable for certain types of projects and highlight areas where further research may be needed.[15]

In table 1 summary of the Literature Review is presented .

4. DESIGN

4.1 Proposed System Architecture

A mobile app for attendance management is meant to make the process of keeping track of attendance easier and more efficient. Instead of writing down attendance manually on a paper-based system and then compiling it, there may be misrepresentation of data, so this app provides a user-friendly way for teachers to manage attendance records and students to view when they were absent. For Admin and teacher, the app also shows and manages total attendance of a student or group of students for a day or a session. Login: The student user authenticates their credentials to access the attendance management system. View Attendance: The student can view their attendance records. and teachers and admin can manage the attendance: The student can mark their attendance for a particular session. In the Update Attendance module: The administrator can update attendance records manually as shown in figure 1, if required. Sync Data with Moodle: The system synchronizes attendance data with the Moodle platform in real-time. Generate Reports: The administrator can generate attendance reports for analysis and record-keeping.
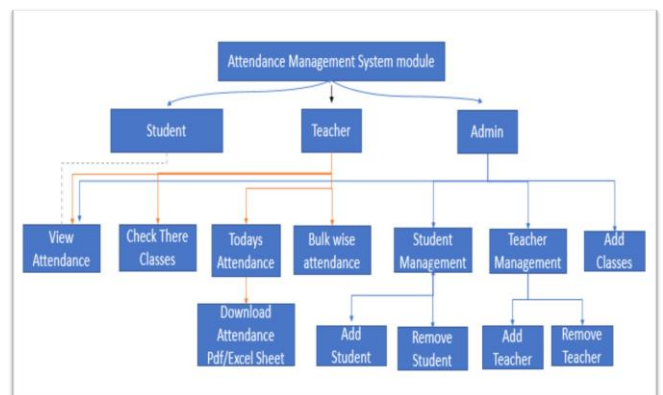
Table 1: Summary of Literature Review

Ref	Year	Title	Methodology	Gap	Publication
1	2021	Framework to Migrate AngularJS Based Legacy Web Application to React Component Architecture	• Choosing the migration strategy • Migration framework	False positive data may be generated	IEEE
2	2021	Comparative study of some applications made in the Vue.js and React.js frameworks	This study used a comparison method by creating two websites, VueJS and ReactJS.	Comparison made by a simple website only for Vue js and react not for any other	IEEE
3	2021	Comparative study of some applications made in the Angular and Vue.js frameworks	This study used a comparison method by creating two Application to do listin angular JS and React JS.	Comparison made by a simple Todo list website only for angular and Vue js not for any other	IEEE
4	2021	Student Attendance System Using An Android Based Mobile Application	It uses geo tag data and it is IoT based.	Student have to register them self for every subject	IEEE
5	2016	Comparative analysis of angular js and react js	Performed a comparison based on the attributes of each framework.	By comparing the size of code manually the result is shown	International Journal of Latest Trends in Engineering and Technology

Figure 1: Proposed approach of AMS

5. METHODOLOGY

5.1 Architecture Design

The research proposes an architecture that comprises the React Native-based mobile application, a server-side application for synchronization the data, and Moodle platform. The architecture allows real-time communication between the Moodle system and mobile application [7], enabling absolute attendance tracking and data updates.

5.2 Implementation Details

The implementation details include the programming languages, development environment, and libraries used in building the React Native application. It also describes the integration steps with Moodle, as shown in figure 2, including the establishment of secure authentication processes, the adoption of efficient data, synchronization techniques, and the utilization of appropriate APIs to facilitate smooth data exchange. when the teacher can be marked the attendance, it will update on Moodle at a time
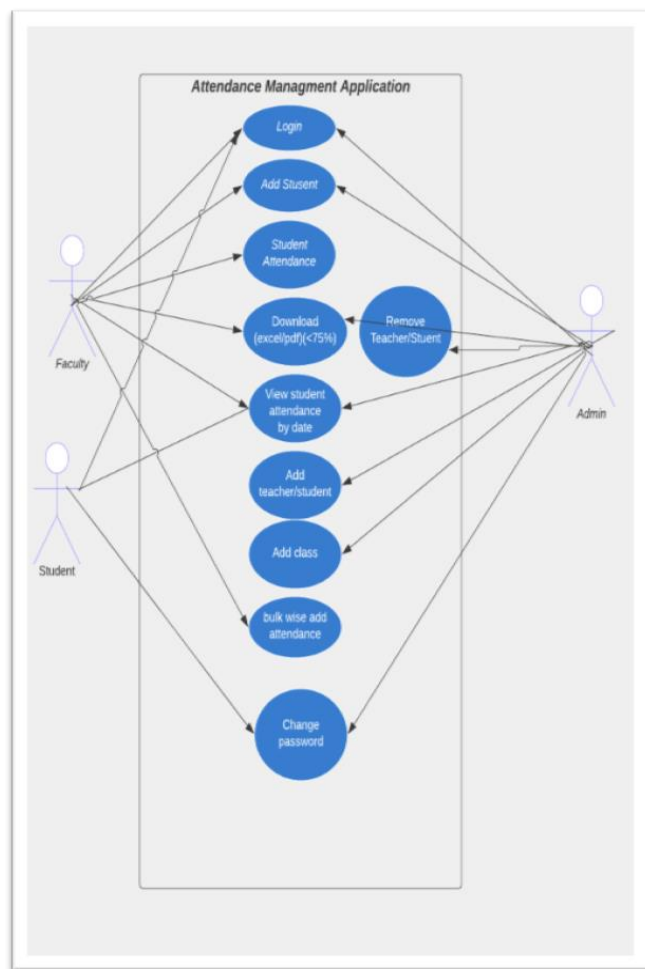

Figure 2: Use Case Diagram of AMS

6. TESTING

• Unit Testing: In unit testing, individual components or units of code are tested to ensure their functionality and correctness.

For example, functions or modules responsible for login, Moodle attendance marking, data update, reflection, report generation, and report download are tested to verify their proper functioning [21].

• Integration Testing: Integration testing verifies the proper integration and communication between different components or modules of the application. It ensures that they work together seamlessly. In this case, the integration between the attendance marking module and Moodle integration would be tested to ensure that attendance data is synchronized correctly.
• Functional Testing: Functional testing validates the functional requirements of the application [18]. It involves testing specific features and functionalities to ensure they work as intended. In this case, functional testing would include scenarios such as marking attendance, generating attendance reports, and downloading attendance records.

• User Interface (UI) Testing: UI testing focuses on the user interface elements of the application to ensure they are visually appealing, responsive, and user-friendly. The user interface of the React Native mobile app would be tested to ensure that teachers can easily navigate, mark attendance, and access attendance reports, while students can view their attendance easily. [8]

• Performance Testing: Performance testing evaluates the responsiveness and scalability of the application under various load conditions. In this case, the performance of attendance marking, report generation, and attendance download functionalities would be tested to ensure they perform well with multiple concurrent users and large amounts of data [19].

• Usability Testing: Usability testing focuses on the overall user experience and ease of use of the application. It involves gathering feedback from actual users, such as faculty members, to assess the application's intuitiveness, efficiency, and overall satisfaction. This helps ensure that the application meets the needs and expectations of its users.

• User Interface (UI) Testing: UI testing focuses on the user interface elements of the application to ensure they are visually appealing, responsive, and user-friendly. You would test the user interface of the mobile app in React Native to ensure that teachers can easily navigate, mark attendance, and access attendance reports. and student can view attendance easily

7. RESULTS AND DISCUSSION

The System is designed in such a way that the login page is to be designed with the user validations as shown in figure 3, username and password to ensure the authentication. If the user does not enter the correct information, the admin, teacher, or student will not be able to login in the application. After login of admin, Admin can view all attendance records of the session as shown in figure 4, admin can add students, teachers and courses, and also can generate reports of attendance as

shown in figure 5. after the login of teachers, teacher can view the list of courses as shown in figure 6 and list of students enrolled in courses as shown in figure 7 and mark the attendance day-wise as shown in figure 8 accordingly and teacher can View the attendance of that session and download the data sheet in the form of pdf and excel format, as shown in figure 9 and after login of student, students can view the attendance as shown in figure 10.
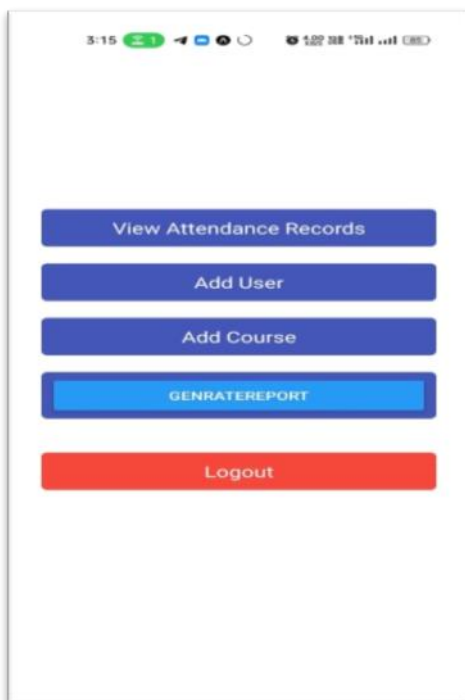

Figure 5: Admin View if admin as a teacher.

Figure 3: Login page.

Figure 6: Teacher View

Figure 4: Admin View

Figure 7: After teacher clicking subjects View
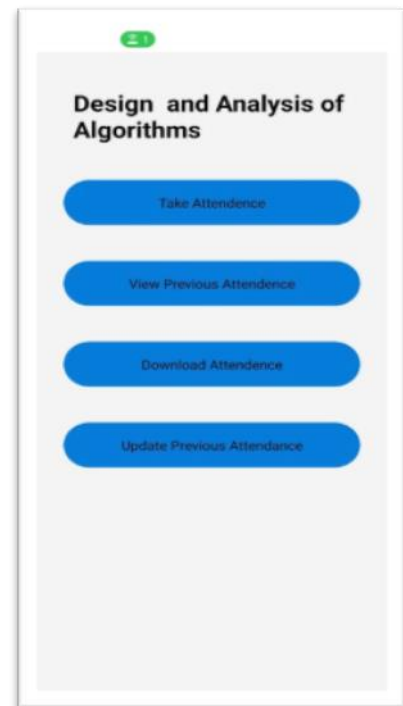

Figure 9: View Attendance

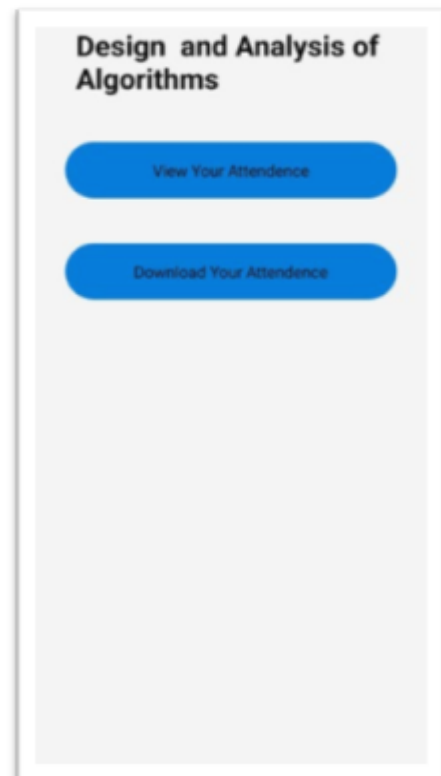**Figure 8:** Teacher Mark attendance

Figure 10:: View the attendance

8. CONCLUSION

The existing manual methods of attendance management were found to be time- consuming and prone to errors. To address these issues, a new solution was proposed in the form of a student attendance system. The proposed system utilizes technology to make attendance management easier and more efficient for both students and educators. With its user-friendly interface, mobile accessibility, and real-time data access, the proposed system has the potential to improve the accuracy and efficiency of attendance management compared to existing methods.

9. FUTURE WORK

The app can be expanded to use biometric of students to mark attendance of students instead of traditional roll call methods. We could also integrate it with results database so that students can view their results directly inside the app. We can send push notifications to every students if there are any changes in schedule of class or any important announcements.

REFERENCES

1. Anurag Kumar and Ravi Kumar Singh **Comparative analysis of angularjs and reactjs**. International Journal of Latest Trends in Engineering and Technology, 7(4), 2016.
2. Nabeel Salih Ali, Ahmed Hazim Alhilali, Hasanein D. Rjeib, Haider Alsharqi, and Basheer Al Sadawi. **Automated attendance management systems: systematic literature review.** International Journal of Technology Enhanced Learning, 14(1):37, 2022.
3. Anik Hanifatul Azizah, Siti Zuliatul Faidah, Muhammad Bahrul Ulum, and Putri Handayani. **Exploration of React Native Framework in designing a Rule-Based Application for healthy lifestyle education.** In 2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI), pages 391–394, Jakarta, Indonesia, October 2021. IEEE.
4. Mar ́ıa Lucia Barr ́on-Estrada, Ram ́on Zatarain-Cabada, Jorge Abraham Romero-Polo, and Julieta Noguez Monroy. Patrony: **A mobile ap- plication for pattern recognition learning.** Education and Information Technologies, pages 1–24, 2022.
5. Alex B ̈uchner. IEEE, 2022.
6. HuaFeng Chen and JunQiao Xiong. **Design and implementation of venue reservation based on react native.** In 2022 International Conference on Artificial Intelligence and Computer Information Technology (AICIT), pages 1–4, 2022.
7. William Danielsson. **React native application development: A comparison between native android and react native**, 2016.
8. Deni Kurniawan Deni and Ferida Yuamita Ferida. **Usability testing penggunaan menu kartu hasil studi di website sistem informasi akademik universitas teknologi yogyakarta.** Jurnal Teknologi dan Manajemen Industri Terapan, 2(I):41–52, 2023.
9. Damiano Distante, Massimo Villa, Nadia Sansone, and Stefano Faralli. Mila: **A scorm compliant interactive learning analytics tool for moodle.** In 2020 IEEE 20th International Conference on Advanced Learning Technologies (ICALT), pages 169–171, 2020.
10. Andre Julian Irawan, Fenina Adline Twince Tobing, and Eunike En- dariahna Surbakti. **Implementation of gamification octalysis method at design and build a react native framework learning application**. In 2021 6th International Conference on New Media Studies (CONMEDIA), pages 118–123, 2021.
11. Shritesh Jamulkar, Milind Kumar Verma, and K. Jairam Naik. **An echo-based autonomous real-time system for efficient maintenance of students attendance.** International Journal of Intelligent Information and Database Systems, 15(4):353, 2022.
12. Vipul Kaushik, Kamali Gupta, and Gupta Deepali. **React native application development.** 03 2019.
13. Vipul Kaushik, Kamali Gupta, and Gupta Deepali. **React native application development.** 03 2020.
14. Cornelia Mihaela Novac, Ovidiu Constantin Novac, Raluca Marina Sferle, Mircea Ioan Gordan, Gyongyi Bujdoso, and Camelia Maria Dindelegan. **Comparative study of some applications made in the Vue.js and React.js frameworks.** In 2021 16th International Conference on Engineering of Modern Electric Systems (EMES), pages 1–4, Oradea, Romania, June 2021. IEEE.
15. Ovidiu Constantin Novac, Damaris Emilia Madar, Cornelia Mihaela Novac, Gyongyi bujdoso, Mihai Oproescu, and Teofil Gal. **Comparative study of some applications made in the Angular and Vue.js frameworks.** In 2021 16th International Conference on Engineering of Modern Electric Systems (EMES), pages 1–4, Oradea, Romania, June 2021. IEEE.
16. Diniz-Junior Raimundo N. V., Caio Cesar L. Figueiredoy, Gilson De S.Russo, Marcos Roberto G. Bahiense-Junior, Arbex Mateus V. L., Lanier M. Dos Santos, Raimundo F. Da Rocha, Renan R. Bezerra, and Felipe T. Giuntini. **Evaluating the performance of web rendering technologies based on JavaScript: Angular, React, and Vue.** In 2022 XVLIII Latin American Computer Conference (CLEI), pages 1–9, Ar- menia, Colombia, October 2022. IEEE.
17. Sami Shaban, Mohi Eldin Magzoub, Margaret Elzubeir, Omar Hilal Shaban, Ahmed R. Alsuwaidi, Mohammed Al Houqani, Alsajir Basheer, Zeeshan Noor Mohammed, Widad El Jaily, and Ahmed Fathy Abdellatif Mohamed. **Developing a student attendance app using QR codes:** educa- tional and practical considerations. International Journal of Technology Enhanced Learning, 13(1):92, 2021.
18. Yan Wang, Lijuan Jia, Hongjian Cao, Ziqi Jing, and Huan Huang. **Applications of cucumber on automated functional simulation testing.** In 2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C), pages 861–862, 2021.
19. Husam Abu Zahra and Samer Zein. **A systematic comparison between flutter and react native from**

automation testing perspective. In 2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), pages 6–12, 2022.

20. Xingwei Zhou, Wenshan Hu, and Guo-Ping Liu. **React-native based mobile app for online experimentation.** In 2020 39th Chinese Control Conference (CCC), pages 4400–4405, 2020.

21. Miroslaw Zielinski and Rix Groenboom. **Using advanced code analysis for boosting unit test creation.** In 2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), pages 279–283, 2021.