International Journal of Computer Applications Technology and Research
Volume 12–Issue 11, 01 – 04, 2023, ISSN:-2319–8656
DOI:10.7753/IJCATR1211.1001

App Sheet for Course Attendance Monitoring in Higher Learning Institutions: A Citizen Developer Approach

Kelvin Rweshobora
Eastern Africa Statistical Training Centre
Dar es salaam, Tanzania

Ali Khelef
Eastern Africa Statistical Training Centre
Dar es salaam, Tanzania

Ramadhani Mrisho
Eastern Africa Statistical Training Centre
Dar es salaam, Tanzania

Abstract: Low-code and no-code development platforms have been around for a long time, but they have only recently become popular. Platforms like Appian and Mendix make users with no coding experience to build complex applications. Citizen developers have been on the rise since the inception of low-code/ no-code app development. However, this phenomenon has not yet benefited learning institutions in Tanzania. Many of the tasks performed by instructors are repetitive and time-consuming, such tasks include recording and tracking student attendance.

A traditional approach to record student attendance is performed by asking every student to sign on an attendance list that passes through all students during the beginning of lectures. This approach is inefficient in term of time and reliability especially in a large class, where a student can sign on the attendance list for other students who are not present in the class. Several automated attendance systems have been proposed based on biometric recognition, barcode, and other technologies such as mobile device. However, all of the previous developed systems require programming skills which sometimes become difficult for Citizen developer to come up with such system.

AppSheet as a no-code development platform provides features that empowers non-developers to customize their applications to suit their specific organizational needs. It enables users to create intelligent actions and interfaces autonomously. Thus, this paper explore the potential of App Sheet by proposing the Attendance Monitoring Application (CAMA) to automate student attendance processes for learning institutions in Tanzania. The results have shown that the CAMA can be used as a solution to assist course managers in registraton and monitoring of students perticipation in the course.

Keywords: App Sheet, low-code, no-code, business process automation, Citizen Developer

1. INTRODUCTION

Low-code and no-code development platforms have been around for a long time, but they have only recently become popular. McLean (2021) defines low/no-code development as a visual software development method that allows developers to build mobile or web applications by dragging and dropping components. In the early 2000s, low-code platforms became more popular, with platforms like Appian and Mendix providing users with a graphical user interface for building applications. These platforms made it possible for users with no coding experience to build complex applications. The COVID-19 pandemic has contributed the rise and awareness of no/low code development platforms. This in turn has paved way for the rise of citizen developers who have proved to be able to create useful and department specific software solutions. Citizen developers have been on the rise since the inception of low-code, no-code app development. However, this phenomenon has not yet benefited learning institutions in Tanzania.

Many of the tasks performed by instuctors are repetitive and time-consuming, such tasks include recording and tracking student attendance. Student attendance is a crucial factor for the performance of the students. In most of the academic institutions, student attendance is taken as one of the requirements for the student to take the exam (Sunaryono et al., 2021). A traditional approach to record student attendance is performed by asking every student to sign on an attendance list that passes through all students during the beginning of lectures. This approach is inefficient in term of time and reliability especially in a large class, where a student can sign on the attendance list for other students who are not present in the class. Several automated attendance systems have been proposed based on biometric recognition, barcode, and other technologies such as mobile device (Mahboob et al., 2016; Sunaryono et al., 2021). However, all of the previous developed systems require programming skills which sometimes become difficult for non-programmer (Citizen developer) to come up with such system.

AppSheet is an application that provides a no-code development platform for application software using data sources like Google Drive and other cloud-based spreadsheet and database platforms(“AppSheet,” 2023). It provides opportunity for citizen developer such as course instructor to develop an application that can automate the student attendance. AppSheet simplifies the development process by allowing simultaneous creation of desktop and mobile applications with minimal coding required. This flexibility empowers non-developers to customize their applications to suit their specific organizational needs. Additionally, AppSheet incorporates artificial intelligence and utilizes Google's hosting language, enabling users to create intelligent actions and interfaces autonomously. Thus, this paper explore the potential of App Sheet by proposing the Attendance Monitorng Application (CAMA) to automate student attendance processes for learning institutions in Tanzania.

2. RELATED WORK

Several organizations have seen the potential for using low-code platforms for web system and app development. Wijesekara et al., (2020) developed an electronic record keeping system at a pediatric clinic in Colombo South Teaching Hospital, Sri Lanka with the aim of obtaining a user friendly electronic DBMS for record keeping, audit and data analysis.The group used googlesheet to create data table as the backend which was then linked to AppSheet for the front end component of the app.

(Martinez & Cisterna, 2023) Petrović et al.,(2020) showcased the application of AppSheet as they demonstrated a rapid development of data driven application using Appsheet and Apps Script. They were able to come up with two artifacts from two use cases, smart home energy management application and COVID-19 patient risk assessment. They have achieved significant development time reduction as compared to traditional app development approach.

The construction industry has also seen the application of low-code in the digitization of its daily processes with the aim of improving the industry.The authors in (Martinez & Cisterna, 2023) discuss the development and deployment of a low-code/AI solution that automates the data processing of paper delivery notes on-site .They observed a significant change in the processing time in comparison to their previous process.The low-code powered process showed a 78% reduction in process time.

The Ponorogo Regency library and Archives Office have also utilized the services of AppSheet as a means to optimize the performance of Archivist Human Resources. From the results AppSheet improved the effectiveness of the performance of archivist human resources in the Office of the Library and Archives of Ponorogo Regency (Sitaviana & Indrahti, 2022).

3. CASE STUDY

Teaching process covers many aspects like attendance tracking, record keeping, continuous assesment and report production. These activities are both tidias and time consuming. with the aid of low-code, this paper aims to empower teachers in every levelof the education system with developers power to automate these tasks. They will assume the role of citizen developers.

This paper showcases a low-code applications Course Attendance Monitoring Application (CAMA) developed using Googles AppSheet and GoogleSheet. This application focuses on course Attendance Monitoring enabling proper record keeping and monitoring of students perticipation.

4. METHODOLOGY

In this context, we have chosen rapid application development (RAD) as the primary approach due to its adaptability in quickly creating an initial working prototype of the application and subsequently refining it continuously.

4.1 APPLICATION ARCHITECTURE

The designing of CAMA involves three google sheet tables for data storage and Appsheet for the UI design figure 1.

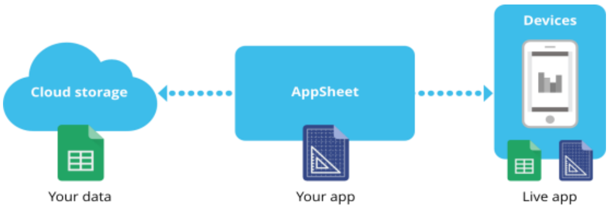


Figure 1: Design Archtecture

Source: Google AppSheet

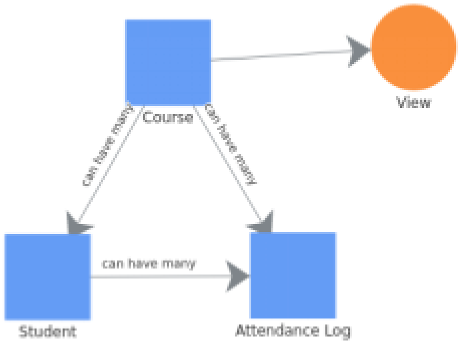


Figure 2: Graph View Presentation

Source: AppSheet

4.2 BACKEND COMPONENT

The backend component used was a google sheet called Coure Manager that has three sheets namely Attendance log, students and course.

Table 1:Google Sheet Tables

SHEET	COLUMN NAMES	SHEET	COLUMN NAMES	SHEET	COLUMN NAMES
Attendance Log	ID	Students	First Name	Course	Course
	Email		Last Name		Description
	Course		Headshot		Image
	Date		Email		
	Time		Course		

Table 1 shows all the sheets for data recording that act as the backend for the application

4.3 FRONTEND COMPONENT

The frontend is the display side of the application where users interact with the app. This works frontend is based on AppSheet. By a single click on the backend interface, an app is created based on the table from Table 1.

Table 2: Frontend Application View

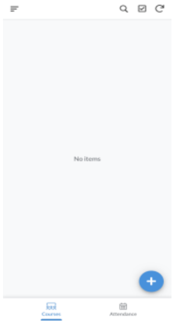
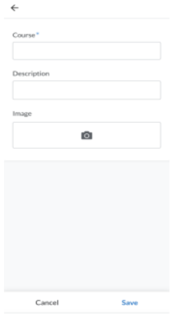
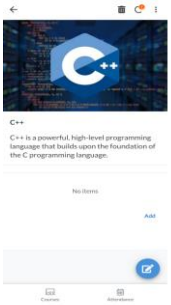
		 C++ C++ is a powerful, high-level programming language that builds upon the foundation of the C programming language. No items Add Courses Attendance
Initial screen	Course Creation	Course

Table 2 shows the app after initializing it, course creation and the course appearance. The app has several controls for adding a course and editing the course details.The app allows the instructor to register a student and assign him/her a course.

Table 3:Students Registration Views

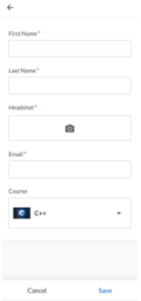
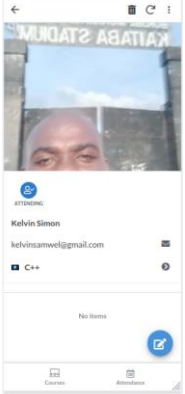
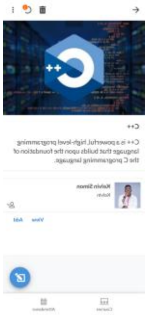
 First Name * Last Name * Headshot * Email * Course C++ Cancel Save	 KATABA STADIUM ATTENDING Kelvin Simon kelvinsamwel@gmail.com C++ No items Courses Attendance	
Student Registration	Students Profile	Course with students

Table 3 shows students registration and how they appear on the course page and their respective profile.

5. CONCLUSION

In conclusion, this Course Attendance Monitorng Application (CAMA) has been developed successfully to assist course managers in registraton and monitoring of students perticipation in the course. This application is developed with minimal cost which uses google spreadsheet as the main database (backend) storage while Appsheet as application development platform(frontend). This application could be accessed by all the academic staff and making the course manager own their course. For further applications, the course management application could be used by any training institution for the same purpse.

6. REFERENCES

AppSheet. (2023). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=AppSheet&oldid=1171803914

Mahboob, T., Irfan, S., & Karamat, A. (2016). A machine learning approach for student assessment in E-learning using Quinlan's C4.5, Naive Bayes and Random Forest algorithms. *2016 19th International Multi-Topic Conference (INMIC)*, 1–8. https://doi.org/10.1109/INMIC.2016.7840094

Martinez, E., & Cisterna, D. (2023). *Using Low-Code and Artificial Intelligence to Support Continuous Improvement in the Construction Industry*. 197–207. https://doi.org/10.24928/2023/0236

Petrović, N., Roblek, V., Radenković, M., & Nejković, V. (2020). *Approach to Rapid Development of Data-Driven Applications for Smart Cities using AppSheet and Apps Script*.

Sitaviana, S., & Indrahti, S. (2022). Optimizing The Performance of Archivist Human Resources Through Appsheet at The Ponorogo Regency Library And Archives Office. *International Journal*, *6*(4).

Sunaryono, D., Siswantoro, J., & Anggoro, R. (2021). An android based course attendance system using face recognition. *Journal of King Saud University - Computer and Information Sciences*, *33*(3), 304–312. https://doi.org/10.1016/j.jksuci.2019.01.006

Wijesekara, D. S., Peiris, P. L. S., Fernando, D. S., Palliyaguru, T. D. N., & Fonseka, W. A. D. N. (2020). Developing an electronic record keeping system at a paediatric clinic in Colombo South Teaching Hospital, Sri Lanka. *Sri Lanka Journal of ChildHealth*,*49*(2),116.https://doi.org/10.4038/sljch.v49i2.8958