

Intelligent ERP System

Dr. C. Sunil Kumar¹, Pullagoru Bhargavi², Ponnamm Ashwith³, Ramantha Karthikeya⁴, S. Aryan Khan⁵

¹Professor, ^{2,3,4,5}Students, Computer Science and Engg., Hyderabad Institute of Technology and Management. Gowdavelly Village, Medchal Mandal, Medchal-Malkajgiri District, Hyderabad, Telangana, India.

ABSTRACT - Educational institutions constantly produce and keep up to date academic and administrative information. Although traditional Enterprise Resource Planning (ERP) systems centralize records such as attendance, examination results, fees, assignments, student profiles and details about staff, many important updates still rely on users logging onto the ERP website or checking their email. The Intelligent ERP System being proposed extends the communication features of an educational ERP by integrating central cloud-based data management with real-time mobile push notifications and Android home screen widgets. Updates made or entered via the ERP can be linked to the appropriate user role so that students, faculty members, parents and administrators get notifications that are relevant to them. Students will receive notifications about attendance, assignments, examinations, results, placements and announcements; faculty members will get notifications regarding their timetable, meetings, departmental matters and the progress of their students; parents will receive alerts about fees, attendance, examinations and student performance; and administrators will be able to send out targeted notices or emergency messages. Firebase Cloud Firestore is suggested for central data storage and Firebase Cloud Messaging is to be used for delivering the notifications. The aim of the system is to cut down on the need for repeatedly visiting the portal, reduce delays in communication and improve access to important academic information.

Keywords - ERP, Notification, Firebase Cloud, Widgets, Communication & Information System.

1. INTRODUCTION

Educational institutions handle huge amount of information on students, faculty, examinations, attendance, fees, assignments and institutional announcements. ERP platforms have become a critical part of this environment, providing a centralized mechanism for storing and accessing academic and administrative records. In education, centralized digital systems are used, as in the case of existing solutions such as SAP Campus Management, Oracle Campus Solutions, Digii, Moodle and Blackboard.

Although centralized ERP platforms can facilitate information management, communication may not necessarily match data management. Users may need to log in more than once for new information to be

recognized. Academic email messages may be read late, SMS communication can introduce recurring costs, and general messaging applications can mix academic messages with personal conversations. Thus, an update can exist in the institutional system without having the intended stakeholder at the right time. It addresses this communication barrier by adding a real-time notification layer to the ERP workflow. Mobile notifications can be sent for relevant changes instead of just portal visits. Android widgets offer another way to access information by displaying selected information right on the device's home screen.

Android widgets give us another way to get information. They show you the info, on your phone's home screen. This method is not meant to replace. To add to the ERP data-management functions. ERP data-management functions stay the same. Now you can see key data without opening the main system.

2. PROBLEM STATEMENT

The main problem is not a lack of data. The problem lies in the lag between when information's updated and when the stakeholder becomes aware of it. Current ERP environments keep records in a central place yet communication is usually spread over portals, email, SMS and messaging apps. This dependence on checks can lead to missed attendance alerts, missed exam schedules, missed fee reminders, missed assignments and missed institutional notices.

This paper identifies a research gap. Educational ERP systems can be strengthened by adding real-time role-specific notification delivery to the existing information-management layer. Firebase Cloud Messaging and Android notification facilities offer technologies, for this communication requirement.

3. OBJECTIVES

1. Improve communication between the institution and its stakeholders with real-time mobile push notifications so that people get updates instantly.
2. Provide role-based communication delivery for students, faculty members, parents and administrators so that each group gets the information.

3. Deliver academic and administrative updates without needing repeated ERP portal access so that communication stays smooth and efficient.
4. Use cloud storage to support timely access to institutional information ensuring that everyone can find what they need quickly and keep communication flowing.
5. Provide Android home screen widgets for visibility of selected information so that users can see key updates at a glance and stay in communication.
6. Reduce communication delays and manual effort while improving the user experience giving everyone a better interaction, with the institution.

4. PROPOSED METHODOLOGY

The system uses a central cloud workflow. All academic and admin information goes into the ERP interface. Is then saved in the Firebase Cloud Fire store. When something changes the system creates a notification that matches the type of update and the role of the person who should receive it. Firebase Cloud Messaging then sends the notification to the registered device. The overall process can be summarized as:

ERP Update → Cloud Data Store → Role Identification → Notification Generation → Mobile Delivery → Widget / App View



Figure 1: Centralized architecture for the proposed Intelligent ERP System

The architecture shown in Fig. 1 Includes an ERP interface a cloud data layer, a notification service and role-based end users. The administrator is responsible, for providing updates. Firestore is used to store information in a centralized way. The notification layer figures out who should receive the updates and sends them to the devices. This setup keeps data management and communication as connected parts.

```

erp pseudocode
BEGIN IntelligentERPSystem
  INPUT userID, userRole, updateType, updateData
  // Authenticate the user
  authStatus = AUTHENTICATE_USER(userID)
  IF authStatus == "SUCCESS" THEN
    // Check the role of the authenticated user
    IF userRole == "ADMINISTRATOR" THEN
      // Validate the information entered/updated by admin
      validationStatus = VALIDATE_UPDATE(updateData)
      IF validationStatus == "VALID" THEN
        // Store data in cloud database (Firestore)
        storeStatus = STORE_DATA_IN_FIRESTORE(updateType, updateData)
        IF storeStatus == "SUCCESS" THEN
          // Identify the recipient role based on update type
          recipientRole = IDENTIFY_RECIPIENT_ROLE(updateType)
          recipientList = GET_USERS_BY_ROLE(recipientRole)
          // Generate notification content
          notificationTitle = GENERATE_NOTIFICATION_TITLE(updateType)
          notificationMessage = GENERATE_NOTIFICATION_MESSAGE(updateData)
          // Send notification via Firebase Cloud Messaging
          notificationStatus = SEND_PUSH_NOTIFICATION(
            recipientList, notificationTitle, notificationMessage
          )
          // Update Android widget if applicable
          IF IS_WIDGET_SUPPORTED(updateType) == TRUE THEN
            UPDATE_ANDROID_WIDGET(recipientList, updateData)
          ENDIF
          // Check notification delivery status
          IF notificationStatus == "SUCCESS" THEN
            LOG_NOTIFICATION_STATUS(updateType, "DELIVERED")
            OUTPUT "Information successfully delivered to intended users."
          ELSE
            RETRY_NOTIFICATION(
              recipientList,
              notificationTitle,
              notificationMessage
            )
            LOG_NOTIFICATION_STATUS(updateType, "RETRY_REQUIRED")
            OUTPUT "Notification delivery retry initiated."
          ENDIF
        ELSE
          LOG_UPDATE_STATUS(updateType, "STORAGE_FAILED")
          OUTPUT "Update failed to store information."
        ENDIF
      ELSE
        LOG_UPDATE_STATUS(updateType, "VALIDATION_FAILED")
        OUTPUT "Update failed: Invalid information."
      ENDIF
    ELSE
      // Fetch role-based information for the user
      permittedData = GET_ROLE_BASED_INFORMATION(userID, userRole)
      IF permittedData IS NOT EMPTY THEN
        DISPLAY_INFORMATION(userID, permittedData)
        OUTPUT "Relevant information displayed successfully."
      ELSE
        OUTPUT "No information available for the selected role."
      ENDIF
    ENDIF
  ELSE
    OUTPUT "Access denied: User authentication failed."
  ENDIF
END IntelligentERPSystem
  
```

Figure 2: Overall Pseudocode of the Proposed Intelligent ERP System

The code in Figure 2 shows the steps of the new system. It includes checking data saving it in the cloud finding out who is involved creating alerts and sending them to the people using a phone app or a small tool, on the screen.

4.1 Role-Based Information Delivery

Role	Important Updates	Delivery Purpose
Students	Attendance, assignments, examinations, results, placements, announcements	Timely academic awareness
Faculty	Timetable changes, meetings, department updates, student progress	Operational coordination
Parents	Fees, attendance, examinations, performance	Student monitoring and alerts
Administrators	Notices and emergency alerts	Targeted institutional communication

4.2 Notification Workflow

1. The administrator updates administrative details that shape the learning environment ensuring every change is intentional and purposeful.
2. The updated information is stored in the cloud database, where the updated information remains safe and accessible, for anyone who needs it.
3. The system identifies the stakeholder role linked to the update. Makes sure the right people see the data so no one gets lost in the shuffle.
4. A push notification event is generated for the intended recipients letting them know something new has arrived and keeping everyone in the loop.
5. The notification is delivered to the registered device. Appears right when the user checks their phone making it quick and easy to stay informed.
6. Selected information can also be displayed through an Android home screen widget giving access at a glance and saving time on the go.

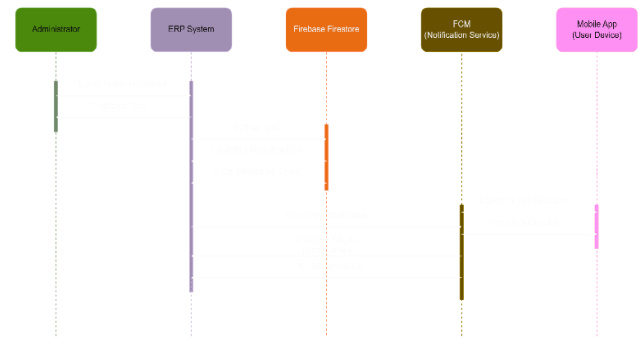


Figure 3: Notification workflow of the proposed Intelligent ERP System.

In Fig. 3 The notification workflow explains how an institutional update is handled and sent to users using Firebase Cloud Messaging. The workflow also includes how the widget is updated and how delivery status is tracked.

5. IMPLEMENTATION ENVIRONMENT

The Intelligent ERP System that we propose is an educational management solution that will be used by institutions. It is meant to help students, faculty, parents and administrators talk to each other by using a platform. The way the Intelligent ERP System will run uses data storage, mobile app building. Push notification tools to give everyone quick access, to school information.

COMPONENT	TECHNOLOGY/ PLATFORM	PURPOSE/ DESCRIPTION
Development Platform	Android	Mobile application development
Cloud Database	Firebase Cloud Firestore	Storage and synchronization of ERP data
Notification Service	Firebase Cloud Messaging (FCM)	Push notification delivery
Mobile Interface	Android UI	Displaying academic and administrative information
Home Screen Feature	Android App Widgets	Quick access to selected updates
User Roles	Student, Faculty, Parent, Administrator	Role-based information delivery
Data Management	Firebase	Cloud-based institutional data handling

The proposed system includes mobile application development, cloud-based data storage push notification

delivery user interface design and Android widget support. The proposed system uses Firebase Cloud Firestore to store and sync data in one location. Firebase Cloud Messaging, also known as FCM is used to send push notifications to devices that are registered with the proposed system. The Android application layer handles the interface. Also supports a home-screen widget.

The Android application layer can connect to the cloud database and the notification service. The proposed system also makes sure that users only see the information they are allowed to access based on their roles.

5. SYSTEM MODELING

5.1 Overview

The Smart ERP System is created as a platform that connects information, from the organization with the people who need it. The Smart ERP System model shows how the administrator, the ERP system the cloud database, the notification service and the mobile application work together. The Smart ERP System handles updates and then sends the right information depending on what each user does. System modeling, which is part of the Smart ERP System gives a picture of how the system works and how the different parts talk to each other. To do this the Smart ERP System uses Unified Modeling Language (UML) models to show how users interact and the order in which the system parts communicate.

5.2 Use Case Model

The use case model shows how the major stakeholders interact with the Intelligent ERP System. The main actors are the Administrator, Student, Faculty and Parent. The Administrator handles data like attendance, exams, assignments, fees, results and announcements. Students can view their details. Get important alerts. Faculty members receive updates about timetables, meetings, department tasks and student information. Parents get updates on attendance, fees, exams and academic progress.

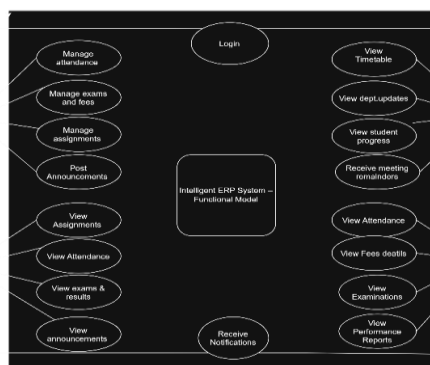


Figure 4: Use Case Model of the intelligent ERP System

In Fig. 4 the use case is described and it shows how different people involved will interact with the ERP system. It shows that what information people get and what messages they receive depends on the job they have. This way of working makes sure that people only see information that's relevant, to them. It also helps make communication clearer and more direct.

5.3 System Components and Data Flow

The proposed system has four parts : the ERP interface, the cloud data-management layer, the notification layer and the mobile presentation layer. The ERP interface is where people enter and manage information about the institution. Firebase Cloud Firestore is the part that stores the records. Firebase Cloud Messaging sends push notifications to devices that are signed up. The mobile app and Android widgets are the parts that users see and use to get and look at information. All these parts work together to create a flow of information from the institution, to the people who need it (shown in the Fig 5)



Figure 5: Data flow of the intelligent ERP System

6. RESULTS AND DISCUSSION

The finalized project proposal shows that the Intelligent ERP System will make institutional communication faster and easier to access. The main benefit of the Intelligent ERP System is that users will need to visit the portal often

and will see information that is clearly separated by role.

The proposed workflow gives a way to check how well notifications are sent whether role-based settings are correct how visible widgets are and how reliable the system is while it is being installed. I believe that numbers about delivery time, notification success rate, user engagement or how much manual checking is cut should only appear after tests. No made-up numbers appear in this paper.

The design is very useful for schools and universities where attendance problems, exam dates, assignment dates, fee reminders and other announcements need to be handled quickly. By linking the ERP information with mobile notifications the Intelligent ERP System will turn existing data into useful actions, for its users.

7. IMPLEMENTATION

Stage	Activity
Requirement analysis	Study existing ERP functions, users and communication requirements.
Architecture design	Define the ERP, cloud database, notification and mobile presentation layers.
Cloud integration	Configure Firebase services for centralized storage and notification support.
Mobile integration	Implement notification handling and Android widget presentation.
Testing	Verify role-based delivery, update handling, notification reliability and usability.
Deployment	Prepare the system for practical institutional use after successful testing.

8. FUTURE SCOPE

The proposed architecture has the potential to grow with advanced analytics better notification priority rules, support for more mobile platforms interfaces in multiple languages, stronger controls for user preferences and connections, with additional institutional ERP systems. Future work could also look into analytics that measure how effective notifications are and how to deliver information in a personalized way. These improvements should be looked at after the main notification system and data processes are fully built and tested.

9. CONCLUSION

The Smart ERP System is suggested as an improvement to educational ERP communication. Current ERP systems do

a job of bringing together academic and administrative information but people might still have to look at portals or other ways to find important updates. The idea is to link data storage with mobile push alerts and Android tools so that the right information can get to students, teachers, parents and school leaders faster.

The plan is about making communication cutting down on work that people have to do themselves and making things easier to get to. It does not aim to replace the ERP system. Firebase Cloud Fire store and Firebase Cloud Messaging are the base for the cloud and message part and Android tools are another way to get to information quickly. This work shows a way to build a better system for educational information. Results, about how it works should come from real testing and using it.

9.1 COMPARISON OF RESULTS

Depending on the user's role, the system offers a variety of functions. The dashboards for Students, Faculty, Parents, and Administrators show access to academic and administrative information that is based on role. The implementation combines the main modules of the proposed Intelligent ERP System into a single mobile interface, thus eliminating the need to consult various sources in order to obtain the relevant information. The results show that the proposed role-based ERP model has been practically implemented. Figure 5, 6, 7, and 8 show this.

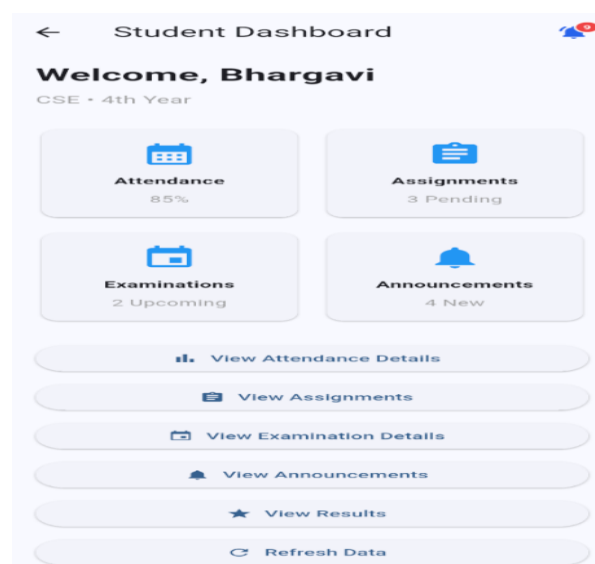


Figure 5: Student Dashboard of the Intelligent ERP System

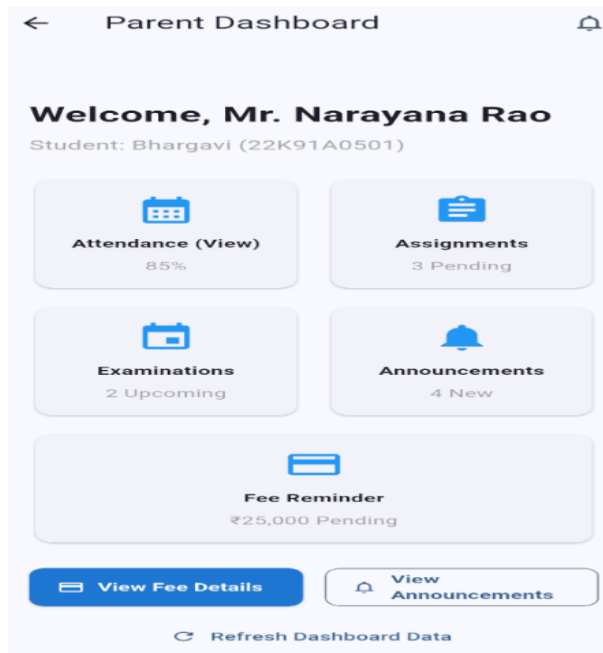


Figure 6: Parent Dashboard of the Intelligent ERP System

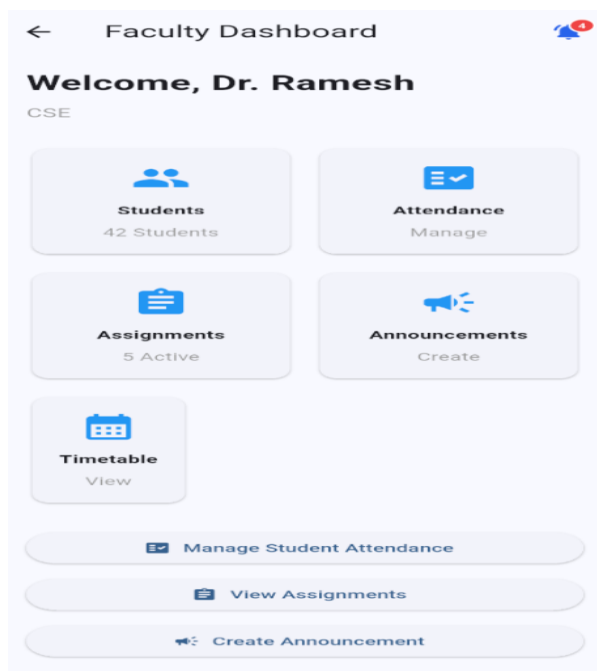


Figure 7: Faculty Dashboard of the Intelligent ERP System

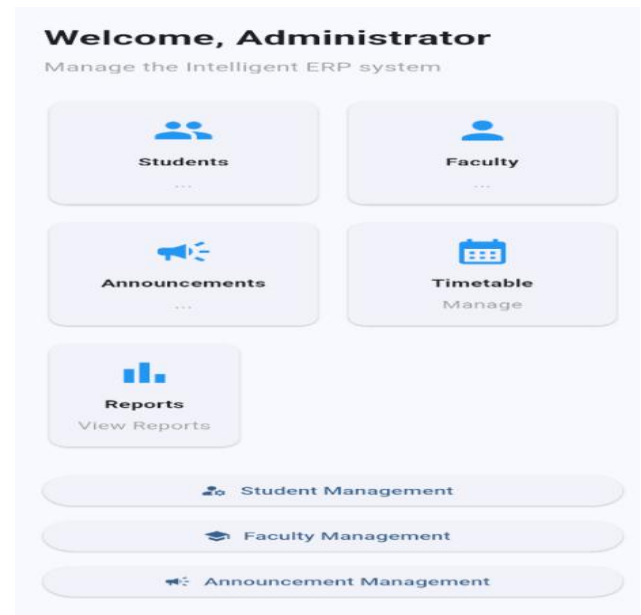


Figure 8: Administrator Dashboard of the Intelligent ERP System

10. REFERENCES

- [1] K. Schwab, SAP Campus Management – Higher Education & Research Solutions, SAP SE, Official Documentation.
- [2] Oracle Corporation, Oracle Campus Solutions Documentation, Oracle Higher Education Solutions, 2024.
- [3] Digii Team, Digii ERP – Smart Campus Management System, Official Product Documentation, 2024.
- [4] M. Dougiamas, Moodle Learning Management System Documentation, Moodle Pty Ltd., 2024.
- [5] Google Firebase Team, Firebase Cloud Messaging (FCM) Documentation, Google Developers. Available at: <https://firebase.google.com/docs/cloud-messaging>.
- [6] Android Developers, Notifications Overview and App Widgets Documentation, Google Developers. Available at: <https://developer.android.com>
- [7] IEEE Xplore Digital Library, Research Papers on Mobile Push Notifications and Educational ERP Systems, IEEE, 2020–2025.
- [8] Google Scholar, Research Articles on ERP Systems, Mobile Notifications, and Educational Information Systems, accessed 2025.