International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Convolinknsphere - A Cross Platform Application for Parent Teacher Communication

Sanskruti Farde, Monali Jadhav, Trupti Jadhav, Rakhi punwatkar

ABSTRACT

There are various challenges related to parent-teacher communication. These challenges are often discussed in educational literature, news articles, forums, and other online resources. Parent-teacher communication issues can arise from factors such as busy schedules, language barriers, differing communication preferences, and the need for timely and effective information sharing. Schools and educators often seek solutions to improve communication with parents and guardians to enhance student support and overall educational experience.

Several available technologies can address parent-teacher communication challenges, such as dedicated communication apps providing messaging, event notifications, and progress sharing. Learning Management Systems offer centralized platforms for resource sharing and progress updates. Email, text messaging, and social media platforms can facilitate quick updates and engagement. Video conferencing tools enable virtual meetings, and student information systems provide parent portals for attendance and grade tracking. Choosing a suitable combination of these technologies can create a comprehensive solution for effective parent-teacher communication.

The use of available communication technologies for parent-teacher communication requires a comprehensive approach to address the gaps identified in the literature review papers. Prioritize data security and privacy by ensuring sensitive information is kept confidential. Seamless communication with existing school systems such as student databases is essential to avoid redundancies and ensure accurate information sharing. To overcome language barriers, technology should also facilitate cross-language communication. Enable real-time updates on student progress to complement multimedia content sharing and increase engagement. In addition, addressing potential issues related to technology training for teachers and parents will improve its implementation. Finally, the technology should be scalable and adaptable to evolving communication needs considering the potential challenges outlined in the literature review.

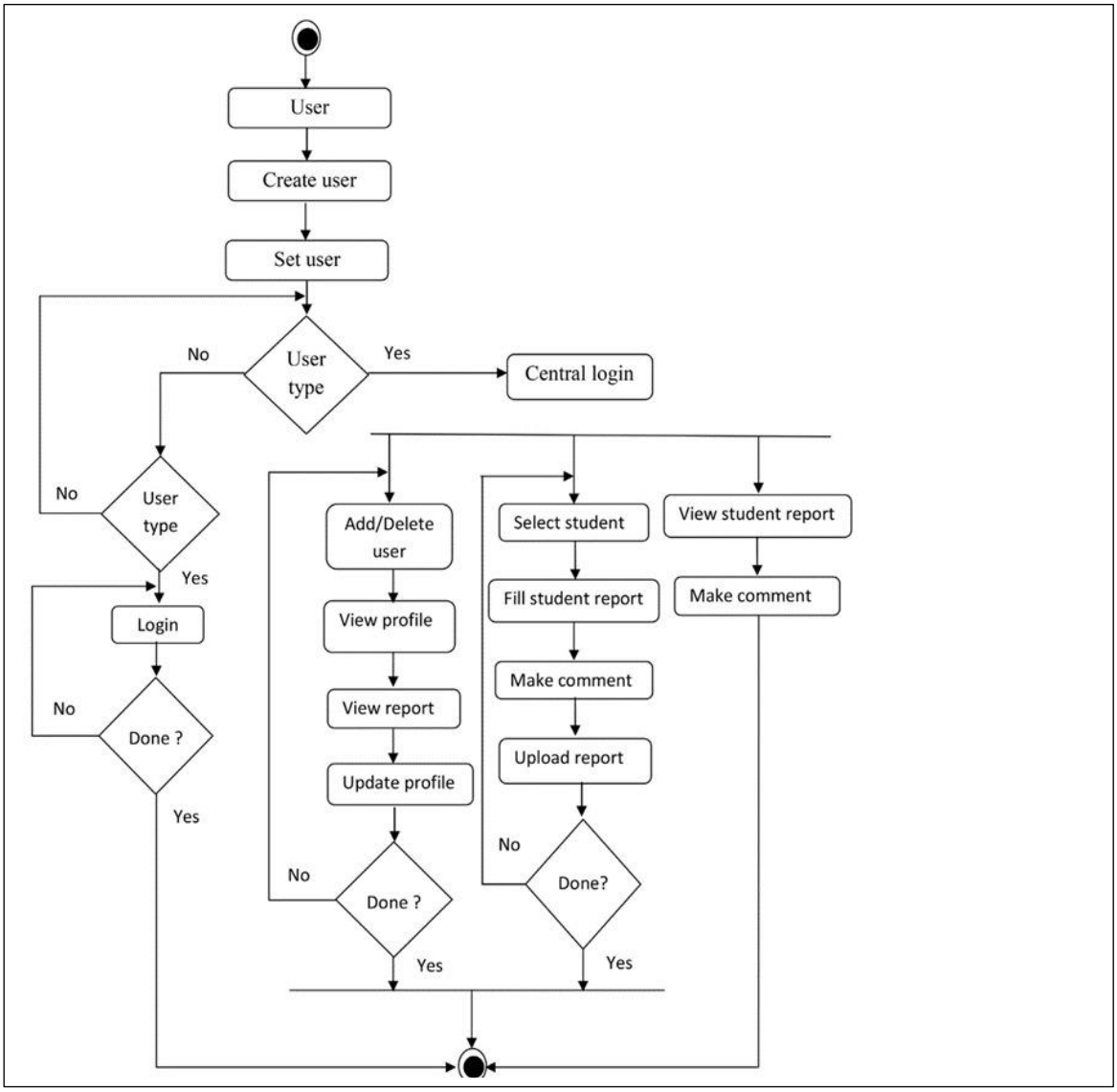
1. INTRODUCTION

The requirements for an effective parent-teacher communication solution include a user-friendly platform with real-time updates on student progress and activities. It should ensure data security, offer customization for communication preferences, and support multilingual communication to bridge language barriers. Integration with existing systems and comprehensive technical support are essential for seamless adoption and scalability.[2]

The challenges in parent-teacher communication are multifaceted. Conventional methods like paper-based communication or face-to-face meetings can lead to untimely information sharing, while failing to cater to modern parents' digital preferences can result in exclusion. Language diversity might be inadequately addressed, causing communication gaps. Furthermore, one-way communication channels limit parent interaction and engagement, potentially creating a passive experience. The security of digital platforms can raise privacy concerns, risking unauthorized access to sensitive student data. Inconsistencies in information from various sources can lead to confusion, and approaches lacking multimedia elements may struggle to capture attention effectively. Technical barriers, especially among lower-income families, can hinder access to communication tools. The perceived additional workload for teachers and resistance to change from both educators and parents further complicate matters. These challenges underscore the need for innovative solutions for development of effective parent-teacher communication.[3]

To address the limitations inherent in current parent-teacher communication methods, suitable remedies emerge in the form of innovative solutions. These encompass the implementation of interactive digital platforms designed to foster real-time engagement, multimedia exchange, and dynamic two-way communication. This technological integration stands poised to amplify parental participation in their children's educational journey. Moreover, harnessing the potential of AI, the development of automated multilingual translation tools emerges as a transformative avenue. By seamlessly transcending language barriers through AI-driven translation, effective communication becomes attainable across diverse parent cohorts. Augmenting these advancements, the provision of comprehensive training initiatives emerges as an indispensable facet. The equipping of both educators and parents with a profound understanding of communication tools engenders proficient utilization, thereby maximizing engagement and realizing the full potential of these progressive strategies.[4]

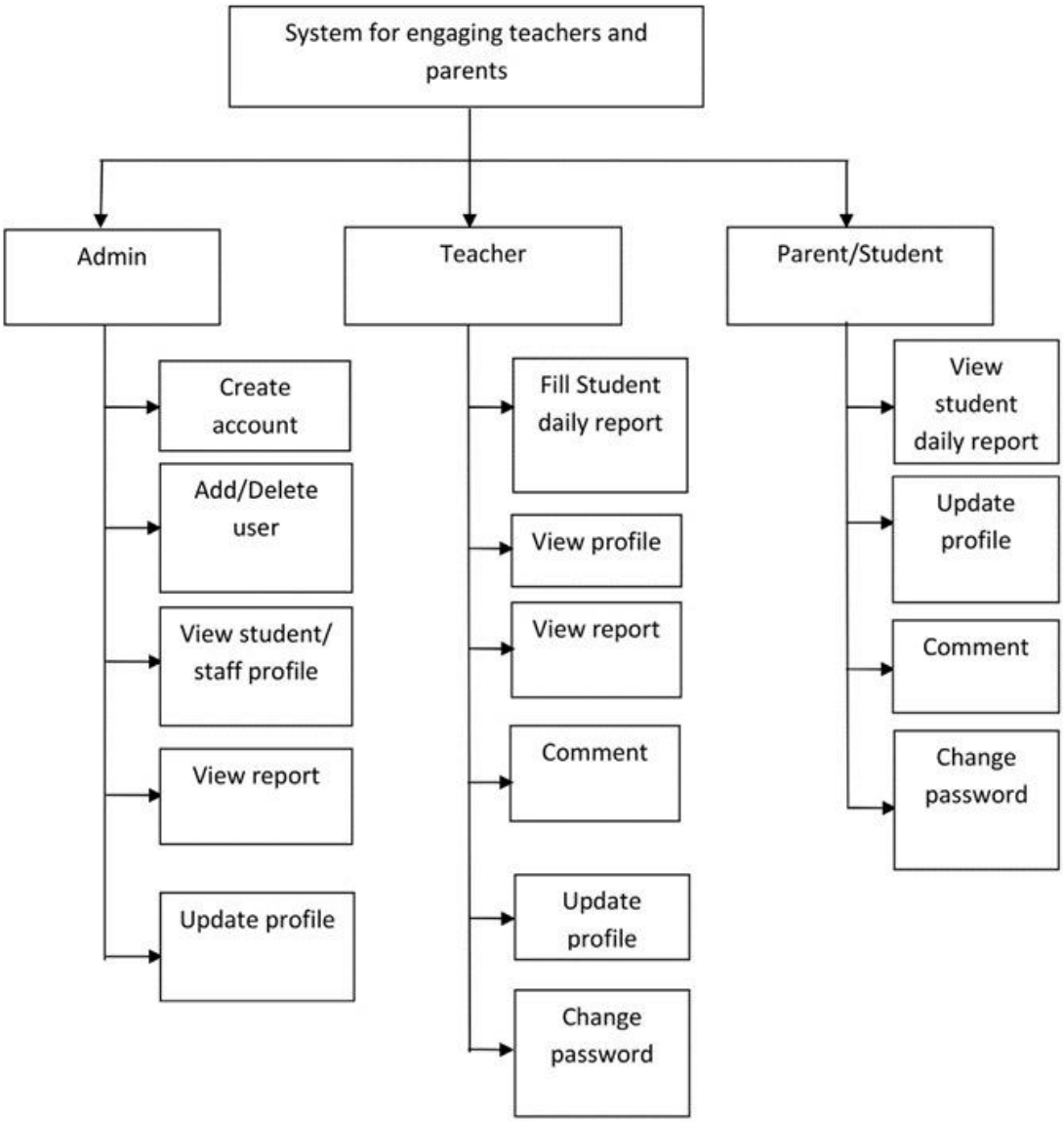
Solution to improve parent-teacher communication is the integration of chatbot and database systems. A personalized chatbot can provide real-time updates on student progress, activities, and upcoming events, as well as address frequently asked questions. The database component optimizes student records, communication history, and predictions, and enables teachers to access relevant information interactively. This dynamic network facilitates communication, provides immediate support, and encourages informed and highly coordinated parent-teacher partnerships.[2]



1.1 MOTIVATION

The eagerness to implement a parent-teacher communication project, such as implementing a chatbot, stems from recognizing the transformative potential of technology in enhancing educational interactions. By introducing a chatbot, schools aim to create a streamlined and efficient communication channel that operates 24/7. This technology-driven solution promises real-time responses to parents queries, freeing up educators time for more personalized interactions. The enthusiasm arises from the prospect of improving parental engagement, bridging communication gaps, and providing parents with a convenient platform to stay informed about their child's progress, events, and activities. The adoption of tools like chatbots aligns with the evolving educational landscape, commitment to digitally-savvy parents and enhancing overall educational experiences.

1.2 PROBLEM DEFITION



1.2 System Behaviour

2 LITERATURE REVIEW

Table 1: Literature Survey

Publisher	Year	Author	Title	Objective
IEEE	2019	Nor-Syahidatul N.Ismail, Muhanmad Nazrin Kamaruddin, Syazana Syahirah Jamaluddin, Nor Saradatul Akmara	Parent Teacher Assisting Monitoring Application System(PTAMA)	The objective of this paper is to Develop an application for parent teacher communication and monitoring of students, main moto of this application are attendance monitoring homework monitoring, announcements through app.
IEEE	2022	Černák, P.Sitiarik and M. Rojček	School web portal as a means of parent-teacher communication	This research examines Estonian primary school teachers' and parents' views on digital communication's effects on child development, covering socialization and academics.
Educational Technology research and developmet	2022	Chen, Y., et	Data Security and Privacy in Database-Driven Parent-Teacher Communication Systems.	To Provide data security in parent-teacher communication systems, emphasizing tech's role in education and discussing its applications and challenges.
Research gate	2023	Surigao Del Norte State University	Enhancing Parental Engagement via the Grade Viewer Application	The Grade Viewer Application is a digital platform that provides real-time access to students' academic progress, attendance, and assignments, fostering seamless communication between parents, teachers, and students
Commons.und.edu	2020	Jaun Wang	Parent Teacher Interaction in Parenting Education	to examine and assess the role and impact of effective communication and collaboration between parents and teachers in the context of parenting education, with a focus on enhancing children's development and educational outcomes

Software Requirement Specifcation

3.1 Introduction

3.1.1 Project Scope

The project aims to create a comprehensive mobile application that enables effective communication between parents and teachers. The application will allow parents to stay updated on their child's academic performance, attendance, and school events. Additionally, teachers can use the platform to share student progress reports, schedules, and important announcements with parents. The application will include features such as real-time messaging, event scheduling, push notifications, and secure data storage. The scope also encompasses the development of an intuitive user interface for both parents and teachers, ensuring a seamless experience across different devices and operating systems.

3.1.2 User Classes and Characteristics

3.1.2.1 Parents

- Parents, as primary users, will utilize the application to stay informed about their child's academic journey.

- They may have varying degrees of technological expertise, necessitating a user-friendly interface that requires minimal learning curve.

- The application should provide parents with real-time updates about their child's academic performance, attendance records, and any other relevant information.

- Secure communication channels must be established to ensure the confidentiality and privacy of sensitive information shared between parents and teachers.

- Compatibility across various devices, including smartphones and tablets running on Android and iOS, is crucial to accommodate different user preferences.

3.1.2.2 Teachers

- Teachers will rely on the application to streamline communication with parents and provide them with timely updates on their child's progress.

- The interface should allow teachers to share comprehensive student reports, attendance records, and behavioural assessments with parents.

- The application should include features that enable teachers to schedule parent-teacher meetings, send timely notifications, and receive feedback from parents.

- User-friendliness is essential, catering to teachers with varying levels of technical proficiency and ensuring a smooth experience across different operating systems.

3.1.3 Assumptions and Dependencies

3.1.3.1 Assumption

End-users have access to compatible Android or iOS devices with sufficient processing power and memory to run the application smoothly.

- A stable and reliable internet connection is available to ensure seamless communication, data synchronization, and real-time updates.

- Users will be responsible for safeguarding their login credentials and taking necessary precautions to prevent unauthorized access or data breaches.

- The development process will adhere to the latest stable version of the Flutter framework and relevant plugins to leverage the most up-to-date features and functionalities.

3.1.3.2 Dependencies

The application will be dependent on a robust and secure cloud-based database system, such as Firebase or a similar service, to store and manage user data securely.

- The performance of the application may be influenced by the quality and consistency of the user's internet connection, emphasizing the need for efficient data handling and synchronization protocols.

- Adherence to the guidelines and policies set by the Google Play Store and Apple App Store is crucial to ensure a seamless application submission, review, and approval process.

- To guarantee the security of user data and communication, the application will implement robust encryption techniques and authentication protocols, safeguarding against potential security threats and breaches.

This detailed specification should serve as a comprehensive guideline for the development of the parent-teacher communication application, offering valuable insights into the project's objectives, user requirements, and critical assumptions and dependencies. Adjust and supplement the document as needed to meet specific project requirements and objectives.

3.2 Functional Requirements

3.2.1 System Feature 1(Functional Requirement)

User Registration

Description: Parents and teachers can create accounts by providing a valid email address and password. They must verify their email addresses during registration.

3.2.2 System Feature 1(Functional Requirement)

User Login

Description: Registered users can log in with their email and password. Authentication should be secure and ensure that users are who they claim to be.

3.2.3 System Feature 1(Functional Requirement)

Password Reset: Users can reset their password by entering their email address and receiving a password reset link via email. This feature helps users regain access to their accounts if they forget their passwords.

3.2.4 System Feature 1(Functional Requirement)

Profile Management: Users can manage their profiles, including updating their names, profile pictures, and contact details. This ensures that user information is always up-to-date.

3.2.5 System Feature 1(Functional Requirement)

Role-Based Access: The system assigns different roles (parent or teacher) to users, each with specific permissions and access levels. Teachers, for example, may have the ability to post announcements and update student grades, while parents may have view-only access.

3.2.6 System Feature 1(Functional Requirement)

Real-time Messaging: Parents and teachers can send and receive real-time text messages within the application. Messages should be delivered promptly to facilitate timely communication.

3.2.7 System Feature 1(Functional Requirement)

Push Notifications: Users receive push notifications for new messages and important updates, even when the app is not in use. Push notifications ensure that users stay informed.

3.3 External Interface Requirements (If Any)

3.3.1 User Interfaces

The application should have an intuitive and user-friendly interface for both parents and teachers, allowing easy navigation and access to various features.

- A visually appealing and responsive design that adapts to different screen sizes and resolutions of Android and iOS devices.

- Clear and concise displays of student performance, attendance, and upcoming events for parents.

- A dashboard for teachers to manage student information, send updates, and schedule events effectively.

- Interactive messaging interfaces for real-time communication between parents and teachers.

- Support for multiple languages and accessibility features to ensure inclusivity and accessibility for all users.

3.3.2 Hardware Interfaces

Compatibility with a wide range of Android and iOS devices, including smartphones and tablets, ensuring seamless functionality across different hardware configurations.

- Utilization of device features such as cameras for document scanning, microphone for voice messaging, and push notifications for timely alerts and updates.

- Optimization for various screen sizes and resolutions, providing an optimal viewing experience on devices with different screen dimensions.

3.3.3 Software Interfaces

- Integration with a secure and reliable cloud-based database system, such as Firebase, for seamless data storage, synchronization, and backup.

- Compatibility with the latest versions of operating systems, including Android and iOS, to ensure optimal performance and feature utilization.

- Integration with authentication services to facilitate secure login and data access for both parents and teachers.

- Utilization of Flutter's APIs and third-party libraries for enhanced functionality and improved user experience.

- Compatibility with common software frameworks and tools for development, testing, and deployment, ensuring smooth collaboration and integration within the development environment.

3.3.4 Communication Interfaces

- Integration with a secure communication protocol for real-time messaging between parents and teachers, ensuring the confidentiality and privacy of all communication.

- Utilization of push notification services to deliver timely updates, event reminders, and important announcements to both parents and teachers.

- Seamless integration with email services to facilitate communication through email notifications and alerts.

- Compatibility with standard communication protocols for data transmission, ensuring reliable and secure data exchange between the application and the cloud-based database.

3.4 Non-functional Requirements

3.4.1 Performance Requirements

System Responsiveness: The system responsiveness requirement focuses on the application's ability to quickly respond to user interactions. It ensures that the app provides a smooth and responsive user experience. For example, when a user sends a message or accesses their student's grades, the system should respond promptly, with low latency, to avoid frustrating users with delays.

Scalability: Scalability is essential as the user base may grow over time. The application should handle an increasing number of users without significant performance degradation. This means the system should be designed to accommodate higher loads, whether it's an increase in user registrations, messaging traffic, or data storage.

Data Loading Speed: This requirement specifies that data, such as student progress reports and messages, should load quickly. Users should not experience significant delays when accessing their data. Fast data loading contributes to a positive user experience and ensures that users can access information efficiently.

Offline Mode: The offline mode requirement ensures that the application offers limited functionality even when the user is not connected to the internet. Users should be able to read previously loaded messages and access certain data while offline. Any actions performed in offline mode, such as composing messages, should synchronize with the server when the user regains an internet connection. This feature is essential for ensuring usability in low or no connectivity areas.

3.4.2 Safety Requirements

Data Privacy: Data privacy is crucial, especially when dealing with student and parent information. The requirement emphasizes that the application should prioritize user data privacy. It should adhere to data protection regulations, such as GDPR, and ensure that user data is not shared with third parties without explicit consent. Additionally, users should have control over their data and who can access it.

Data Backup and Recovery: This requirement mandates that the system must have mechanisms in place for regular data backups and recovery procedures. These measures ensure that data can be restored in case of data loss due to unforeseen events or system failures.

User Authentication Security: Security in user authentication is paramount. The application must implement strong authentication measures, such as encryption and account lockout mechanisms, to protect user accounts from unauthorized access. This helps safeguard user information and prevents potential breaches due to weak or compromised passwords.

3.4.3 Security Requirements

Data Encryption: Data encryption is vital to protect data both in transit and at rest. Sensitive information, such as messages and student records, should be securely encrypted. Encryption ensures that even if unauthorized access occurs, the data remains unreadable and confidential.

Secure API Endpoints: Secure API endpoints are necessary for data exchange. These endpoints should require proper authentication and authorization mechanisms to prevent unauthorized access. This helps maintain data integrity and security during data transfers.

Authentication Logging: Authentication logging is important for security and accountability. The system should log and monitor all authentication and authorization activities, including login attempts and permission changes. These logs can help identify suspicious activities and potential security breaches.

Vulnerability Scanning: Regular security scans and penetration tests should be conducted to identify and address potential vulnerabilities within the application. This proactive approach helps maintain the application's security by identifying and patching security weaknesses before they can be exploited.

3.4.4 Software Quality Attributes:

Usability: Usability focuses on the user experience. The application should be user-friendly, with an intuitive and easily navigable interface. User testing and feedback should be used to continually improve usability, ensuring that users find the application easy to use and navigate.

Reliability: Reliability is essential to ensure the application has minimal downtime and system failures. It should be highly dependable so that parents and teachers can rely on it for effective communication.

Performance Monitoring: Performance monitoring tools should track server response times, database performance, and other key metrics. This helps maintain the system's efficiency and ensures that it continues to perform optimally. It also aids in identifying performance bottlenecks and areas for improvement.

Scalability Testing: Scalability testing ensures that the application can handle increased loads, especially during peak usage times, without significant performance degradation. It helps prepare the application for growth and high-demand periods, such as parent-teacher conference seasons.

Maintainability: Maintainability ensures that the codebase is well-documented and follows best coding practices. It is important for long-term success, as it enables developers to easily maintain and update the application. Regular updates and bug fixes should be carried out to keep the application in good working order.

Availability: Availability indicates how often the application is accessible to users. The application should be highly available, with minimal scheduled downtime for maintenance. Any maintenance activities that may affect system availability should be communicated to users in advance to minimize disruption

3.5 System Requirements

3.5.1 Database Requirements

- A reliable and scalable cloud-based database system, such as Firebase or Firestore, to store and manage user data securely.
- The database should support real-time data synchronization, ensuring that updates made by either parents or teachers are immediately reflected across all devices.
- Secure data encryption and user authentication mechanisms should be implemented to protect sensitive information from unauthorized access and data breaches.
- The database should be capable of handling a large number of concurrent users and data transactions without compromising performance or data integrity.
- Regular data backups and recovery mechanisms should be in place to prevent data loss in case of system failures or unexpected errors.

3.5.2 Software Requirements (Platform Choice)

- Compatibility with the latest versions of Android and iOS operating systems to ensure that the application can run smoothly on a wide range of devices.
- Utilization of the Flutter framework to build a cross-platform application that can deliver a consistent user experience on both Android and iOS devices.
- Integration of platform-specific features and guidelines to ensure that the application adheres to the design and functionality standards of each operating system.
- Utilization of compatible software development kits (SDKs) for Android and iOS to enable seamless integration with the respective platforms and leverage their native capabilities.
- Regular updates and maintenance of the application to ensure compatibility with future releases of the Android and iOS operating systems and the Flutter framework.

3.5.3 Hardware Requirements

- The application should be compatible with a wide range of Android and iOS devices, including smartphones and tablets, to accommodate different user preferences and device capabilities.
- Minimal hardware specifications should be defined to ensure that the application runs smoothly on devices with varying processing power, memory, and storage capacities.
- Integration with device hardware components, such as cameras, microphones, and GPS, to enable features like document scanning, voice messaging, and location-based services within the application.
- Consideration of hardware constraints, such as screen sizes and resolutions, to ensure that the application's user interface remains consistent and functional across different device types and form factors.

3.6 Analysis Models: SDLC Model to be applied

For the development of a parent-teacher communication application using Flutter, the Agile software development lifecycle (SDLC) model is recommended. Agile is well-suited for projects that require frequent feedback, continuous iterations, and the ability to adapt to changing requirements. It emphasizes collaboration, flexibility, and customer satisfaction. Here's how the Agile SDLC model can be applied to the development of the parent-teacher communication application:

3.6.1. Planning: During the planning phase, the project requirements, goals, and priorities are defined. This includes creating a product backlog that outlines the features and functionalities of the application, as well as prioritizing tasks based on the needs of both parents and teachers.

3.6.2. Development: The development phase involves creating small, incremental deliverables called "sprints." Each sprint typically lasts 1-4 weeks and focuses on implementing specific features or functionalities. The development team uses Flutter to build and test these features, ensuring that they meet the requirements and expectations outlined in the product backlog.

3.6.3 Testing: Continuous testing is a crucial aspect of the Agile model. Testing is performed throughout the development process to identify and address any issues or bugs promptly. This ensures that the application functions as intended and provides a seamless user experience for both parents and teachers.

3.6.4 Review and Feedback: At the end of each sprint, there is a review session where stakeholders, including parents and teachers, provide feedback on the features implemented during that sprint. This feedback is incorporated into the next sprint, allowing for continuous improvements and adjustments based on the evolving needs of the end-users.

3.6.5. Iterative Development: The Agile model promotes iterative development, allowing the team to continuously enhance the application based on user feedback and changing requirements. This iterative approach ensures that the application remains adaptable and responsive to the dynamic nature of parent-teacher communication needs.

3.6.6. Deployment: As each sprint is completed, the features are integrated into the application, and the application is tested for overall functionality and compatibility. Regular deployments and updates are made to ensure that the latest features and improvements are available to users.

3.6.7. Maintenance and Updates: Even after the initial deployment, the Agile model encourages ongoing maintenance and updates to address any issues, add new features, and enhance the overall user experience. This ensures that the parent-teacher communication application remains relevant and effective in facilitating communication and collaboration between parents and teachers.

3.7 System Implementation Plan

1. Project Kickoff:

Objective: Start the implementation process with a clear understanding of the project's scope, objectives, and constraints.

Tasks: Gather the project team, including developers, designers, and quality assurance personnel.
Review the project requirements, including functional and non-functional requirements.
Establish project timelines and milestones.
Create a detailed project plan that includes tasks, responsibilities, and deadlines.

2. Environment Setup:
Objective: Set up the development and testing environments to ensure a smooth development process.
Tasks:
Install and configure Flutter development tools and SDKs.
Set up version control systems (e.g., Git) and repositories.
Configure the development IDEs (e.g., Android Studio, Visual Studio Code).
Establish continuous integration and continuous deployment (CI/CD) pipelines.

3. Database Design and Setup:
Objective: Design the database structure and set up the database system to store application data.
Tasks:
Define the database schema, including tables, relationships, and data types.
Set up the database server and create the necessary databases.
Implement data migration scripts to initialize and update the database structure.

4. User Authentication and Authorization:
Objective: Implement secure user authentication and authorization mechanisms.
Tasks:
Develop user registration and login screens.
Integrate third-party authentication providers (e.g., Google Sign-In, Facebook Login).
Implement role-based access control for parents and teachers.

5. Core Features Development:
Objective: Develop the core features of the application according to the functional requirements.
Tasks:
Create user profiles, including profile picture management.
Implement real-time messaging functionality.
Develop the announcement system for teachers.
Build the student progress tracking and grade submission features.
Implement the shared calendar and event reminders.
Develop the file sharing and file download capabilities.
Integrate multilingual support and language selection features.
Implement search functionality for messages and data.

6. Security and Data Privacy:
Objective: Ensure data security and privacy features are implemented correctly.
Tasks:
Implement data encryption for data at rest and data in transit.
Set up secure API endpoints for data exchange.
Log and monitor authentication and authorization activities.

Conduct vulnerability scans and penetration tests to identify and address potential security issues.

7. User Experience and Design:
Objective: Create an intuitive and user-friendly interface for the application.
Tasks:
Design the user interface (UI) with a focus on usability.
Incorporate a consistent color scheme, typography, and branding.
Conduct user testing to gather feedback and make iterative improvements.

8. Quality Assurance and Testing:
Objective: Ensure the application functions correctly and meets quality standards.
Tasks:
Develop and execute test cases to validate the application's functionality.
Perform performance testing to verify speed and responsiveness.
Verify compliance with non-functional requirements, such as security and data privacy.
Address and resolve bugs and issues identified during testing.

9. Documentation:
Objective: Create comprehensive documentation for users, administrators, and developers.
Tasks:
Prepare user guides and help resources.
Document the system architecture, database schema, and APIs.
Create an FAQ section to address common user queries.

10. Deployment and Release:
Objective: Deploy the application to app stores and make it available for users.
Tasks:
Package the application for Android and iOS platforms.
Submit the application to the Google Play Store and Apple App Store.
Configure app store listings with descriptions, screenshots, and promotional materials.

11. User Training and Support:
Objective: Provide users with the resources they need to use the application effectively.
Tasks:
Develop training materials and resources.
Offer customer support and channels for user inquiries and assistance.

12. Marketing and Adoption:
Objective: Promote the application to increase user adoption.
Tasks:
Develop marketing materials, such as promotional videos and social media content.
Create a marketing strategy to reach parents and teachers.
Monitor user adoption and gather feedback for further improvements.

13. Ongoing Maintenance:
Objective: Ensure the application remains functional and up-to-date.
Tasks:
Provide ongoing support to address issues and bug fixes.
Release updates and improvements based on user feedback.
Conduct regular system maintenance and updates.

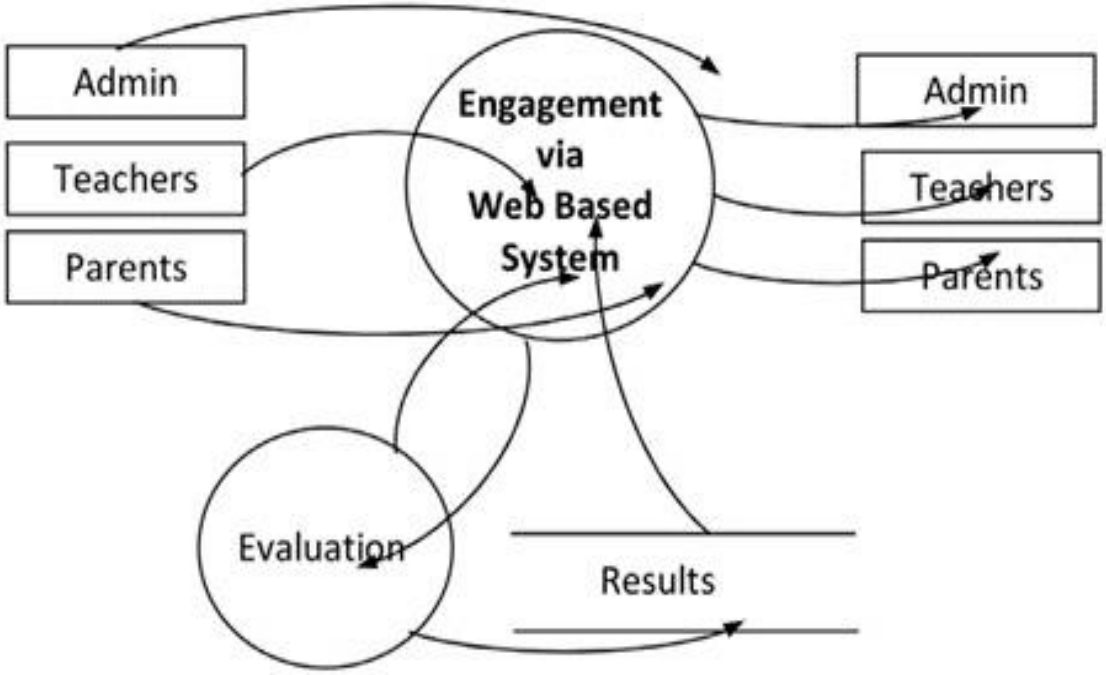
4. System Design

4.1 System Architecture

The parent-teacher communication application will utilize a client-server architecture model. The client side will be developed as a mobile application using the Flutter framework, ensuring cross-platform compatibility for both Android and iOS devices. The server side will be implemented using a scalable and secure cloud-based database system, such as Firebase, which will handle data storage, synchronization, and user authentication. The application will communicate with the server through secure APIs, ensuring the seamless transfer of data and facilitating real-time communication between parents and teachers.

4.2 Data Flow Diagrams

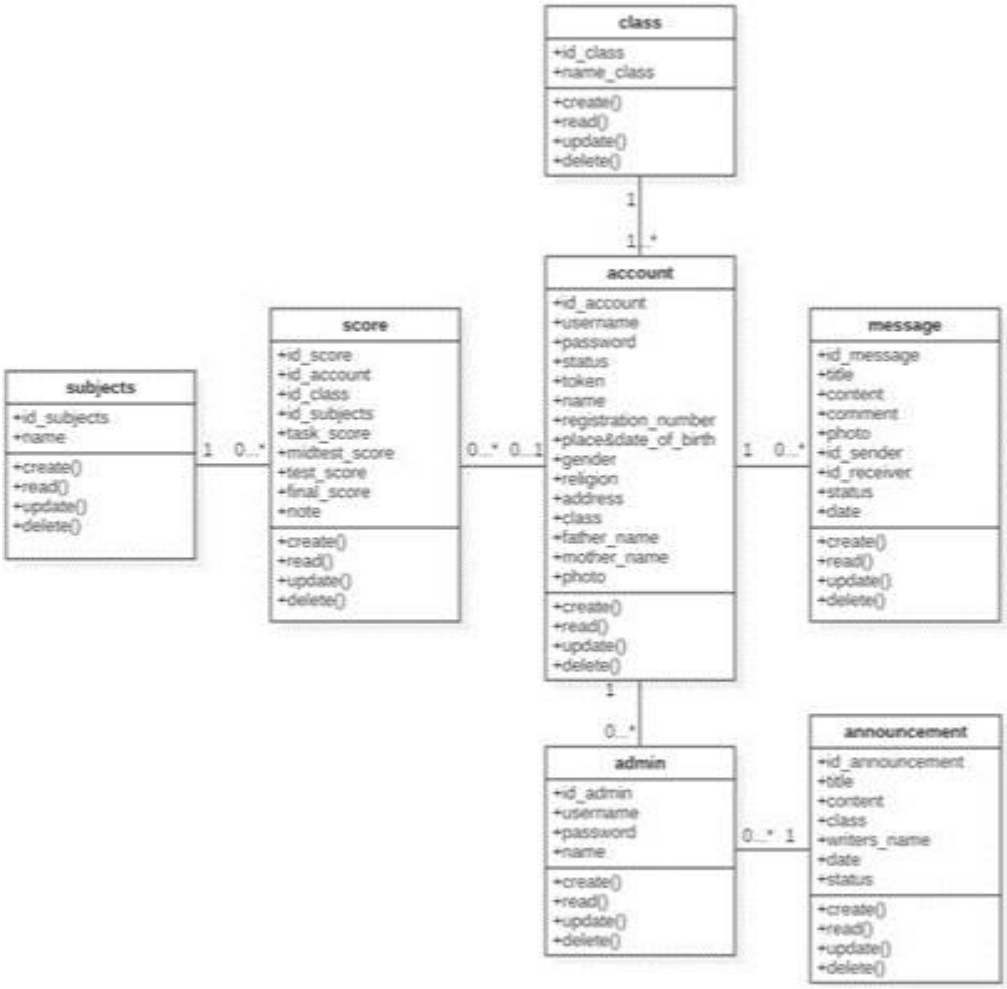
The Data Flow Diagram (DFD) will visually represent the flow of data within the parent-teacher communication application. It will illustrate the data inputs, processes, data storage, and outputs involved in the communication process between parents, teachers, and the database system. The DFD will depict the interactions between different entities, including user authentication, message exchanges, data retrieval, and event scheduling, providing a comprehensive overview of the data flow and processes within the application.



4.1 data flow Diagram

4.3 Entity Relationship Diagrams

The Entity Relationship Diagram (ERD) will serve as a visual representation of the database schema for the parent-teacher communication application. It will define the various entities involved, such as parents, teachers, students, messages, schedules, and events, along with their respective attributes. The ERD will illustrate the relationships between these entities, including one-to-one, one-to-many, and many-to-many relationships, ensuring a clear understanding of the database structure and the connections between different data entities within the application.



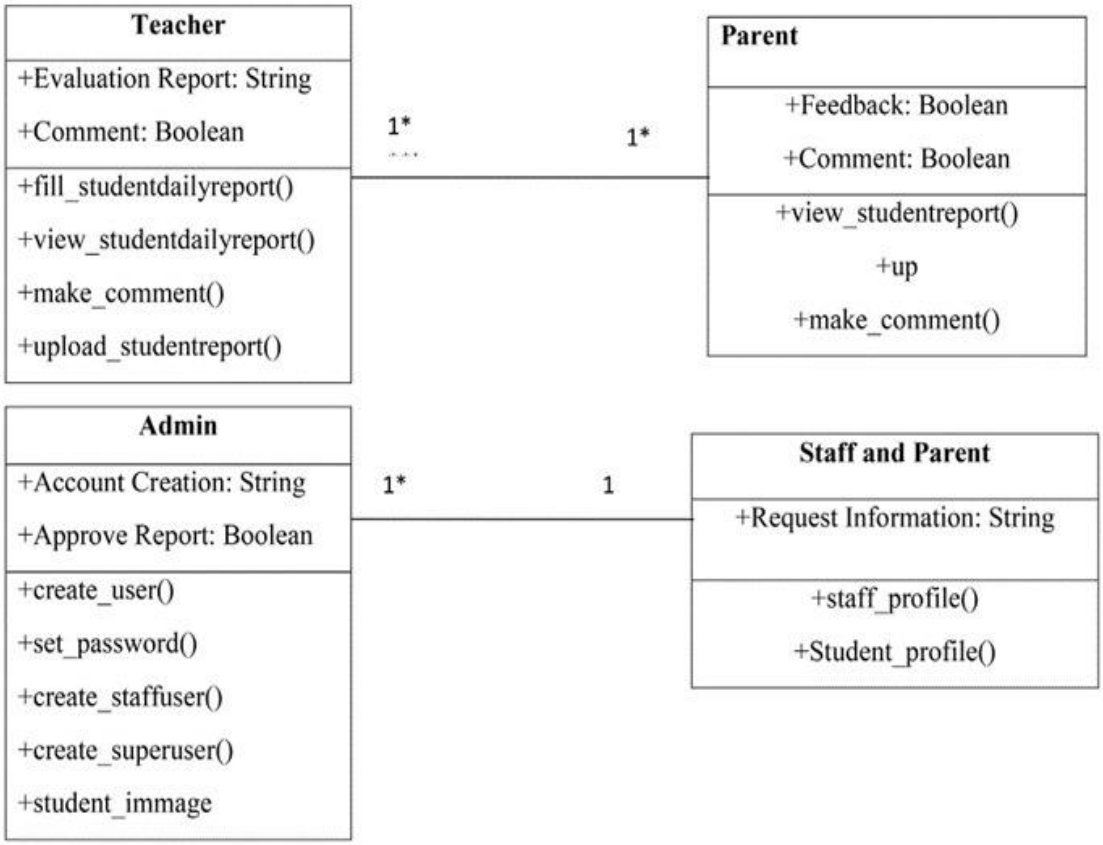
4.2 ER Diagram

4.4 UML Diagrams

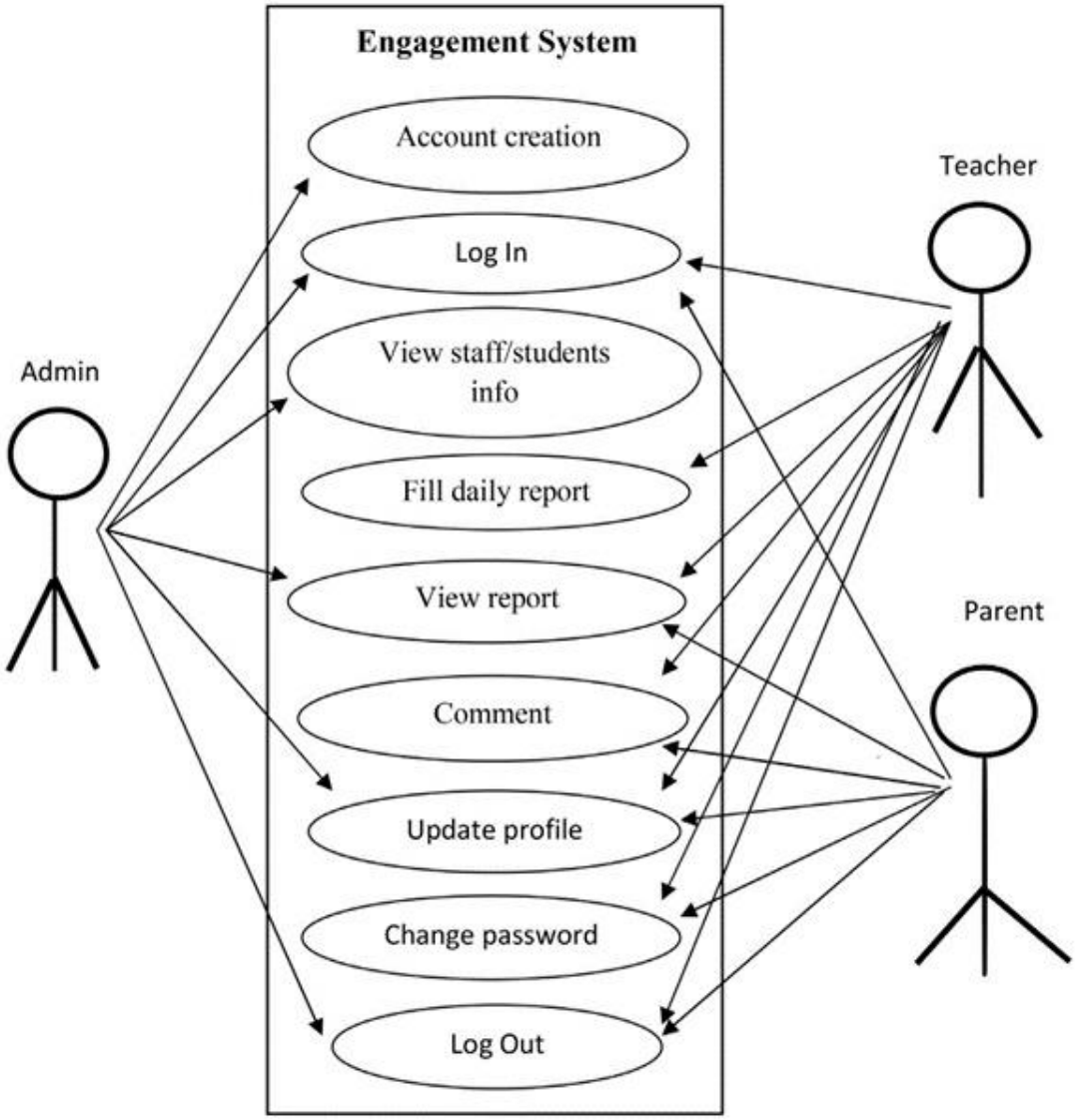
The Unified Modeling Language (UML) diagrams will provide detailed insights into the different aspects of the parent-teacher communication application:

- Use Case Diagrams: Illustrating the various use cases and functionalities of the application from the perspectives of both parents and teachers, including features such as messaging, event scheduling, and data retrieval.
- Class Diagrams: Describing the static structure of the application, including the classes, attributes, and methods associated with parent and teacher profiles, student information, messaging functionalities, and event scheduling.
- Sequence Diagrams: Demonstrating the interactions and message flows between different components of the application, showcasing the sequence of events involved in tasks such as sending messages, scheduling events, and retrieving data from the database.
- Activity Diagrams: Displaying the workflow and activities involved in various processes within the

application, including the step-by-step actions of users when sending messages, accessing student information, and scheduling events.



4.3 Class Diagram



4.1 UseCase Diagram

5 Other Specification

5.1 Advantages

Enhanced Parent-Teacher Communication: The application streamlines communication between parents and teachers, making it easy for them to exchange information, share updates, and discuss student progress. This strengthens the partnership between home and school, benefiting the student's education.

Real-time Updates and Notifications: The app offers real-time messaging and push notifications, ensuring that parents and teachers stay informed about important events, announcements, and student progress. This timely communication keeps everyone in the loop.

Paperless and Environmentally Friendly: By enabling document sharing, grade submissions, and messaging within the app, the need for physical paperwork and printed documents is reduced. This aligns with eco-friendly initiatives and saves resources.

Access Anytime, Anywhere: Parents and teachers can access the application from their mobile devices or web browsers, making it convenient to communicate and stay updated, whether at home or on the go.

Multilingual Support: Multilingual support ensures that users from diverse language backgrounds can use the application effectively. This inclusivity caters to a wider user base.

Improved Student Outcomes: Effective communication between parents and teachers leads to better support for students' educational needs. As a result, the application can contribute to improved student outcomes and academic success.

5.2 Limitations:

Digital Literacy and Accessibility: The application assumes users have access to smartphones and the internet, which may not be the case for all parents. Ensuring accessibility for users with limited digital literacy or access is a challenge.

Data Security Concerns: Storing sensitive student and parent data within the application requires robust security measures to protect against data breaches and unauthorized access. Any security vulnerabilities can pose risks.

Dependency on Technology: The application relies on technology, and any technical issues, such as server downtime or connectivity problems, can disrupt communication between parents and teachers.

Privacy and Data Protection: Balancing the need for real-time communication with the privacy and data protection requirements, especially regarding students, can be challenging. Adherence to privacy regulations is a necessity.

Initial Adoption: Initially, some parents and teachers may be resistant to using the application, either due to unfamiliarity with the technology or a preference for traditional communication methods.

5.3 Applications

K-12 Education: The primary application is within K-12 education systems, where parents, teachers, and students can benefit from streamlined communication, access to student progress reports, and timely information sharing.

Higher Education: Similar applications can be used in higher education institutions to facilitate communication between professors, students, and parents or guardians. They can be valuable for sharing course information and student performance updates.

Language Learning Centers: Language learning centers can use such applications to keep parents informed about their child's progress and any language-specific programs or activities.

After-School Programs: After-school programs, clubs, and extracurricular activities can adopt the app to keep parents in the loop about events, schedules, and their child's participation.

Special Education: In special education, where close collaboration between parents and educators is crucial, these applications can help tailor educational plans to individual student needs and track progress effectively.

Tutoring and Coaching Services: Tutoring services and coaches can use this technology to communicate with parents and provide updates on the student's performance, areas of improvement, and goals.

6 Conclusions & Future Work

In conclusion, the implementation of the parent-teacher communication project, incorporating a chatbot and maintaining a comprehensive student database, represents a pivotal step towards creating a more connected and informed educational ecosystem. By leveraging technology to bridge communication gaps, the project strives to enhance parent engagement, streamline information dissemination, and promote data-driven decision-making. The integration of a chatbot offers personalized interactions, enabling parents to receive timely updates and tailored responses, fostering a stronger partnership between home and school. Simultaneously, the student database empowers educators to gain insights into student performance, enabling targeted interventions for improved learning outcomes. As technology continues to shape the education

7 .REFERENCES

[1] Černák, P. Sitiarik and M. Rojček, "School web portal as a means of parent-teacher communication," 2022 20th International Conference on Emerging eLearning Technologies and Applications (ICETA), Stary Smokovec, Slovakia, 2022, pp. 91-96, DOI: 10.1109/ICETA57911.2022.9974731.

[2] Chen, Y., et al. (2022). Data Security and Privacy in Database-Driven Parent-Teacher Communication Systems. Educational Technology Research and Development, 70(3), 1053-1068 DOI: 10.17148/IJARCCE.2022.11478.

[3] Johnson, C., & Williams, D. (2020). Leveraging Student Databases for Enhanced Parent-Teacher Communication. Educational Communication and Society, 23(4), 465-480.

[4] Lee, J., & Kim, S. (2022). Enhancing School-Parent Communication through Database-Integrated Platforms. Educational Technology and Society, 25(1), 215-228.

[5] Smith, M., et al. (2019). Enhancing Parent-Teacher Engagement through Student Databases. Journal of Educational Computing Research, 54(3), 302-316.

[6] N. -S. N. Ismail, M. N. Kamaruddin, S. S. Jamaluddin and N. S. Akmar, "Parents-Teacher Assisting Monitoring Application (PTAMA) System," 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET), Shah Alam, Malaysia, 2019, pp. 402-407, DOI: 10.1109/ICSEngT.2019.8906298.

[7] T. Haiduwa, O. Mbandeka, P. Kautwima and J. Nelulu, "Enhancing Parent-Teacher Communication through USSD-Based Services amid COVID-19 Lockdown," 2021 IST-Africa Conference (IST-Africa), South Africa, South Africa, 2021, pp. 1-8.

[8] A. Lutfi, F. Saidi and M. Watfa, "A ubiquitous smart educational system: Paving the way for big educational data," 2016 Sixth International Conference on Innovative Computing Technology (INTECH), Dublin, Ireland, 2016, pp. 233-238, doi: 10.1109/INTECH.2016.7845129.

[9] M. Fetaji, B. Fetaji, A. Ajredini and M. Ebibi, "Devising a model of electronic School Management System based on web services for secondary schools in Macedonia," Proceedings of the ITI 2013 35th International Conference on Information Technology Interfaces, Cavtat, Croatia, 2013, pp. 187-192, doi: 10.2498/iti.2013.0573.

8 . PHASES OF IMPLEMENTATION

Sr. No.	Plan Duration	Task to be done	Time Required	Percentage of work done	Status of current phase
1	Aug-Sept	Survey & finding the technology/ database	2 months	5	In process
2	Oct–Nov	Design of Phase1	2 months	0	-
3	Dec–Jan	Implementation of Phase2	2 months	0	-
4	Feb-Mar	Testing and result analysis	2 months	0	-

