International Conference on Science, Engineering
& Management Trends 2026

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

Health AI: Personal Health Advisor Mobile Application for People

Walid Bebal, Joydeb Nandi, Taha Ghole, Prof.S.E.Gawali
Department of Computer Engineering G.M.Vedak Institute Of Technology, Tala Maharashtra, India

Abstract - The rapid evolution of mobile technology has cre- ated opportunities for new types of mobile applications, including applications for providing healthcare information. This paper discusses the design and implementation of a mobile health information software application with a Node.js backend. The primary function of the application is to distribute health-related information via a mobile device. The proposed system adopts a less complex architecture due to the absence of complex backend systems and databases. The system is designed such that the mobile application and backend server communicate via standard HTTP protocols. The proposed system focuses on minimizing the time for implementation and enhancing system responsiveness and simplicity. The results of the evaluation of the system's performance under typical use conditions indicate that the application is responsive and reliable. The developed system demonstrates an efficient use of mobile and server-side technologies for educational and informational healthcare.

Index Terms - Mobile Application, Health Information System, Node.js, Android Application, Software Engineering.

I. INTRODUCTION

The swift development of mobile technologies has pro- foundly altered how information is shared and accessed in a variety of fields, including healthcare. Through mobile appli- cations, users can access health-related information directly on their smartphones in a portable and adaptable manner. Access to healthcare information has never been easier thanks to the widespread use of mobile devices.

Applications for mobile health are crucial for raising user awareness of health issues and offering fundamental advice. Without the need for frequent trips to medical facilities, these applications assist users in understanding general health con- cepts, preventive measures, and lifestyle-related information. Consequently, mobile applications help users become more aware and make better decisions.

Existing healthcare information systems may use complex infrastructures, desktop-based applications, or data storage techniques, making them unsuitable for small-scale appli- cations, especially in academic settings, for delivering in- formation services. This is because mobile applications are considered simple, user-friendly, and promise a better user experience than traditional healthcare systems.

Backend support is significant in the management of communication between mobile applications and the server. Node.js has also gained popularity in the backend environment and has been successful in supporting mobile applications due to the asynchronous and event-driven nature of the Node.js environment. It allows the processing of multiple clients within a short response time.

The project aims to create a mobile health application with a supporting backend developed using the Node.js program- ming language. The developed application provides health information to users via a mobile interface. Additionally, it is designed to be independent of database systems to ensure a simpler architecture. The proposed application shows how to bring mobile programming and server programming together to provide healthcare information in an efficient manner.

II. LITERATURE REVIEW

The adoption of mobile applications in the healthcare do- main has received considerable attention from various re- searchers and developers. Research related to mobile health applications has shown that these applications can function as efficient tools in disseminating health-related information, generating health awareness, and practicing preventive health- care. The mobile devices and their portability make them a conducive environment for information dissemination.

Research conducted by several scholars has already identi- fied the benefits of mobile health applications, demonstrating their potential to overcome traditional health systems like desktop-based applications. Mobile applications offer the abil- ity to use health information at any time and in any location, hence making it more user-friendly and convenient. The im- portance of simple mobile devices in improving usability and user satisfaction was emphasized in the literature.

The technologies used to develop the backend supporting the mobile applications have been discussed extensively by different pieces of research. Node.js is widely used as the backend technology due to its asynchronous and event-driven model, as discussed in the research studies. Multiple requests are handled efficiently by Node.js in supporting the mobile ap- plications, as it provides fast responses to the mobile devices, as mentioned in the research articles.

In some studies, it is necessary to consider the complexity of the healthcare systems that heavily depend on databases and analytics tools. It is essential to point out that such healthcare systems are not necessary for the basic delivery of health information, contrary to the necessities that information systems will be required, especially in informational and academic applications.

Overall, there is considerable support in the literature for developing mobile health information applications utilizing simple backend technologies, which avoid unnecessary com- plexity and utilize core features to deliver reliable and efficient health information services. This project is an extension of such findings by developing a simple mobile health informa- tion application utilizing a Node.js backend.

III. PROBLEM STATEMENT

The proliferation of information over health issues, coupled with the difficulty experienced by users seeking access to these well-structured and reliable pieces of information using mobile devices, forms the major challenge with the current applications, most of which are developed with complexity, high backend processing, and database usage, factors deemed unnecessary for the development and maintenance of health information delivery-based applications.

In some cases, it is noted that the application of mobile health applications requires users to be registered, data to be stored, and synchronized accordingly. This can result in higher resource consumption and decreased system performance, making the deployment process complicated. These features can result in decreased system efficiency rather than usability for academic or small-scale applications.

Another challenge is the absence of mobile health applica- tions that are light and concentrate on information dissemina- tion. Most often, users want to be made aware of health issues and need a guiding application. A simple application with easy access and fewer steps is most appropriate. However, most applications lack this attribute.

Additionally, it is vital for the backend system, which pri- marily deals with the application, to handle multiple requests with efficiency and minimum response time. A conventional server system might prove inefficient in handling the situation without arousing additional complexities.

As a result, the problem being addressed in the proposed solution in the context of the current project is the need for a lightweight mobile health information application that is able to access accurate health-related information in a timely manner without the need for database or other backend services. As a result, the design for a simple mobile platform using a Node.js backend is the desired solution.

Objectives

The principal goal for undertaking this project is to create a mobile health information application, which allows users to access health information in a simple manner. Completing a project in mobile application development involves creating a mobile application that presents health information in a suitable way.

Another important objective to be covered in the mobile ap- plication development plan is to ensure that the user interface of the mobile application is simple and easy to use or intuitive. Care has been taken to highlight ease of navigation with regard to the user interface for an enhanced user experience.

Another objective of the project is to ensure the implemen- tation of backend support using Node.js to enable communi- cation between the mobile app and the backend. The backend is intended to process the requests and send the appropriate responses without the use of storage components.

Maintaining lightweight system architecture is also a sig- nificant goal. This means that the system is easily deployed, managed, and maintained by avoiding backend services and storage. This is specially more important in academic envi- ronments.

Additionally, the project also seeks to assess the response time and stability of the mobile application that will be created. This will further prove that with a simple mobile application and even minimum backend dependencies, efficiency and reliability will be achieved.

Overall, the project aims at showing how effectively the two technologies can be integrated towards developing and design- ing a functional and efficient health information application for mobile devices and server-side systems.

System Overview

Application of Mobile Health Information Application of Mobile Health Information Application Concept
The mobile health information application is an application software de- signed to offer users a chance

to access information concerning health-related affairs through a user-friendly interface. The application uses a very simple design, aiming at content delivery and not content storage.

The system uses a client-server architecture, where the mobile app is a client and communicates with a Node.js backend server. The client sends requests for information related to health, and these requests are handled by the server and then returned to the client. As this system does not make use of a database, all information is statically defined or generated by the server.

The function of the backend server includes routing, re- quest handling, and sending responses. It is managed using Express.js, which handles the application routing. It regulates the flow of the application data from the mobile client to the backend server. It helps the backend effectively handle many requests without any extra system overheads.

The mobile application prioritizes ease of usability and content presentation. The application allows users to interact by employing simple screens that can be used by mobile de- vices. The system's overall design prioritizes ease of usability, responsiveness, and reliability.
A. System Component Overview

Table I
System Components and Description

Component	Description
Mobile Application	Provides a user interface to access health-related information on mobile devices. The application is designed for easy navigation and optimized for mobile screen sizes.
Backend Server	A Node.js-based server that processes incoming requests from the mobile application and delivers appropriate responses without relying on database storage.
Framework	Express.js is used to manage routing, handle HTTP requests, and control the flow of data between the mobile client and the server.
Communication Protocol	HTTP is used for communication between the mobile application and the backend server to ensure reliable request and response handling.

System Architecture
The architecture of the proposed mobile health information application is designed to support efficient interaction between a mobile client and a lightweight backend service. The system adopts a client–server architecture, which separates the mobile user interface from backend processing logic. This separation improves system clarity, maintainability, and performance.

The mobile application serves as the client layer of the sys- tem. It is responsible for displaying health-related information and handling user interactions such as selecting topics and navigating between screens. All actions initiated by the user are processed at the mobile level and forwarded to the backend only when content is requested.
The backend layer is implemented using Node.js with the Express.js framework. It handles incoming requests from the mobile application and routes them to the appropriate service handlers. Since the system does not rely on a database, the backend directly processes requests and delivers predefined or dynamically generated health information without persistent storage.

Data exchange between the mobile application and the backend server is performed using standard HTTP request and response mechanisms. This communication model ensures platform independence and allows the mobile application to operate consistently across different devices and operating systems.

Overall, the system architecture focuses on simplicity, fast response time, and ease of deployment. By avoiding complex backend components and storage mechanisms, the architecture remains suitable for academic use and basic health information delivery through mobile platforms.

Mobile Application System Architecture Diagram

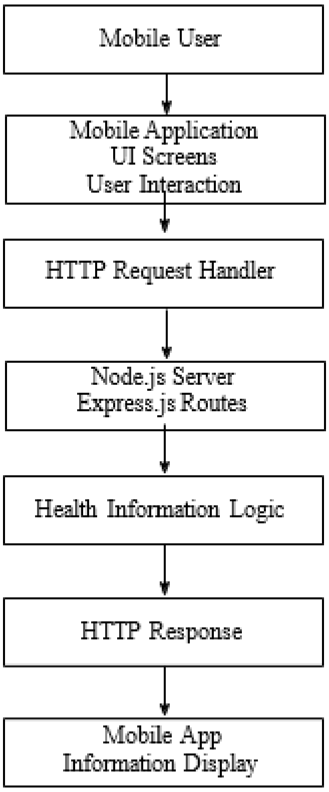

Fig. 1. System Architecture of the Mobile Health Information Application

Requirement Analysis

Requirement analysis is an important phase in the develop- ment of the proposed mobile health information application, as it defines the system functionality and constraints clearly. The main purpose of this phase is to ensure that the application fulfills user needs while maintaining simplicity and efficiency. Since the system is designed as a lightweight mobile applica- tion, the requirements focus primarily on information delivery and user interaction.

The functional requirements describe the basic operations that the system must support. The mobile application should allow users to access health-related information through a simple and intuitive interface. Users should be able to navigate between different health topics smoothly, and the application must be capable of requesting relevant information from the backend server whenever required.

The backend system is required to handle incoming requests from the mobile application efficiently. It should process HTTP requests and return appropriate responses without rely- ing on database storage. This requirement helps reduce system complexity and ensures faster response time during normal usage.

Non-functional requirements focus on the quality attributes of the system. The application should provide quick response time, reliable performance, and ease of use on mobile devices. Additionally, the system should be easy to deploy and maintain due to its lightweight architecture and absence of persistent storage, making it suitable for academic and informational purposes.

Hardware Requirements

The hardware requirements of the proposed mobile health information system are minimal because of its lightweight approach. This is because the system is considered a data delivery system, and there is no need to carry out complicated processing or store data; hence, it does not require extensive resources from a mobile device. The server, at the backend, will require a basic hardware requirement to handle HTTP request handling from the mobile-based system.

Table II
Hardware Requirements

Component	Specification
Mobile Device	Android smartphone capable of running ba- sic applications
Processor	Quad-core or higher (mobile device)
RAM	Minimum 2 GB (mobile device)
Storage	Minimum 100 MB free space for applica- tion installation
Server Machine	Standard PC or laptop for Node.js backend execution
Network	Stable internet connection (Wi-Fi or mobile data)

Software Requirements

The requirements for the software specify the platforms or tools that are necessary for developing or executing the application of mobile health information. The application utilizes commonly used mobile and server-based software, thus ensuring the simplicity of development and execution. The absence of the need to use the database makes the environment simple.

Table III
Software Requirements

Component	Technology
Mobile Operating System	Android OS (version 8.0 or higher)
Mobile Application Type	Native or hybrid mobile application
Backend Runtime	Node.js
Backend Framework	Express.js
Communication Protocol	HTTP
Development Tools	Visual Studio Code, Android Em- ulator
Testing Tools	Web browser, Android Emulator

Implementation Details

Implementation of the proposed mobile health information application was carried out in a simple manner. This appli- cation involves the implementation of a two-layer structure. These two layers are the mobile application layer and the backend server layer. These two layers are separated with the aim of enhancing the clarity of the application.

The mobile application has been created to deliver an intuitive and user-friendly interface suitable and compatible with mobile devices. Users are able to navigate through various health-related categories and request information through sim- ple interaction with the interface. The application is centered around usability and readability to improve user experience for mobile devices.

For the backend server, Node.js is used in combination with the Express.js framework. This server needs to handle the HTTP requests sent by the mobile app and return the required responses accordingly. In the absence of the database, the backend will handle the requests and return responses containing health information accordingly. This will keep the system simple and the response time fast.

The communication between the mobile application and the backend server for exchanging data is done using traditional HTTP request-response methods, which are reliable, platform- independent, and efficient for usage in academic and informa- tional scenarios.

Implementation Components

Table IV
Implementation Components and Description

Component	Description
Mobile Application	Provides the user interface for accessing and navigating health- related information on mobile de- vices.
Backend Server	Node.js-based server that handles requests from the mobile appli- cation and sends appropriate re- sponses.
Framework	Express.js framework used for routing and managing HTTP re- quests efficiently.
Communication Method	HTTP request and response mech- anism used for interaction between mobile app and backend server.
Data Handling	No database used; health infor- mation is served directly through backend logic or predefined con- tent.

ERFORMANCE EVALUATION

In addition, performance evaluation would be carried out to measure efficiency, responsiveness, and stability of the proposed mobile health information application. However, as a lightweight system without dependency on a database, its response time would be considered during its evaluation.

The testing involves requisite HTTP requests being sent to the Node.js backend server by the mobile application. The backend receives the requisite information and proceeds to send data related to health. An analysis of the testing provides an understanding of the performance of the system.

The following tables and graph show the observed perfor- mance of the system.

Response Time Analysis

Table V
Response Time Analysis

Number of Requests	Average Response Time (ms)
1	120
10	150
25	190
50	240

System Stability Evaluation

Table VI
System Stability Evaluation

Test Scenario	System Status
Single Request	Stable
Continuous Requests (10)	Stable
Continuous Requests (25)	Stable
Continuous Requests (50)	Stable

Graphical Performance Analysis

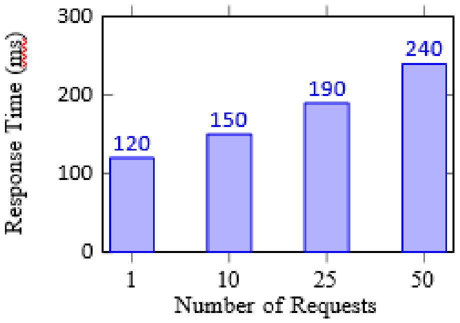

Fig. 2. Response Time vs Number of Requests

This means, by the trend of the results, that with the increase in the number of requests, the application will maintain steady operation and response time within acceptable levels. Since the absence of database operations reduces processing overheads, the response was faster and consistent in performance. The performance results satisfy that this proposed mobile health information application is efficient and suitable for academic and informational use cases.

Limitations

In the proposed application of mobile health information, the features have been achieved as intended; however, the application also has a few disadvantages, which must be taken into consideration. This application is mainly required to deliver information related to health, and personalization is not allowed.

One major drawback of this system, however, is the absence of a database or storage mechanisms. From this, it's evident

that the system lacks the capability to store the user's pref- erences and dynamic health conditions. All the information thus given by the system is either predefined or generated dynamically by the system's backend.

The application does not have the feature of including user authentication and user profiles in the program, and this restricts the possibility of providing the end user with customized services or controlling the level of access to certain services within the program. The reason for choosing to leave out these attributes was to keep the program simple and uncomplicated.

Yet another limitation is that an active internet connection is required for communication between the mobile app and Node.js backend server. This might hinder its use in areas where network connectivity is too poor, especially for use in such environments.

Additionally, the system is only for academic and informa- tional use only. The system does not replace advice, diagnosis, or treatment given by a medical expert. This also points to the possible improvements that can be made to the system in its next releases.

IV. CONCLUSION

The paper presented the design and development of a mobile health information application supported by a Node.js backend. The application provides health-related information through a simple and user-friendly mobile interface. The system will remain lightweight and efficient due to the lack of complex backend services and dependencies of heavy databases.

This implementation is going to show how effectively a mobile application can merge with server-side technologies to provide content in an efficient manner. The performance eval- uation of the results shows fast response time and stability of operation under normal conditions. The proposed client–server architecture clearly separates user interactions from back-end processing.

Overall, the proposed system serves as a useful academic implementation of a mobile application for health information. It demonstrates how it is possible to effectively exploit the Mobile and Backend Technologies for healthcare information delivery and at the same time serves as the easy foundation for future enhancements while keeping things simple and easy to deploy.

future work

While the existing mobile health information application development has given a platform for imparting basic health- related information to users, with regard to the future devel- opment of the application, considerations for incorporating user authentication facilities need to be taken into account to maximize user engagement using the application.

This feature of database integration can be included in the future versions of the application to allow for the dynamic stor- age of contents, preferences, and interaction history of users. This feature would play a crucial role in the management of the contents and the scalability of the system. Alternatively, the application could be provided with the ability to access certain information offline.

Further, the application could be extended to run on multiple mobile platforms and to offer improved user interface designs. Integration with enhanced back-end services, as well as secu- rity systems, could make the applicability of the application even better. Hence, these designs could be very instrumental in transforming the application into an even better mobile health information application.

REFERENCES

1. J. A. A. Akter, S. Ray, and A. Islam, “Mobile applications in healthcare: Opportunities and challenges,” Journal of Health Informatics, vol. 12, no. 2, pp. 45–52, 2018.

2. M. Free, G. Phillips, L. Watson, et al., "The effectiveness of mobile- health technologies," PLOS Medicine, vol. 10, no. 1, pp. 1–9, 2013.
3. D. Luxton, "An introduction to mobile health," in Artificial Intelligence in Behavioral Health, Academic Press, 2016, pp. 21–34.
4. A. Carroll, G. Belanger, "Mobile phone applications and healthcare," Telemedicine Journal, vol. 19, no. 7, pp. 546–551, 2015.
5. K. Mechael, "The case for mHealth in developing countries," Innova- tions, vol. 4, no. 1, pp. 103–118, 2009.
6. S. Patel and A. Desai, "Design considerations for mobile health appli- cations," International Journal of Computer Applications, vol. 120, no. 5, pp. 1–6, 2015.
7. N. Lane et al., "A survey of mobile phone sensing," IEEE Communica- tions Magazine, vol. 48, no. 9, pp. 140–150, 2010.
8. R. Buyya, J. Broberg, and A. Goscinski, Cloud Computing: Principles and Paradigms, Wiley, 2011.
9. S. Tilkov and S. Vinoski, "Node.js: Using JavaScript to build high- performance network programs," IEEE Internet Computing, vol. 14, no. 6, pp. 80–83, 2010.
10. A. Al-Fuqaha et al., "Internet of Things: A survey on enabling technolo- gies," IEEE Communications Surveys, vol. 17, no. 4, pp. 2347–2376, 2015.
11. R. Pressman, Software Engineering Concepts, McGraw-Hill, 2014.
12. S. S. Yau and S. K. S. Gupta, "Middleware for mobile systems," IEEE Internet Computing, vol. 6, no. 4, pp. 52–58, 2002.
13. H. He, "Mobile healthcare systems," IEEE Systems Journal, vol. 8, no. 3, pp. 645–650, 2014.
14. P. Zhang and N. Li, "Usability of mobile information systems," Infor- mation Systems Journal, vol. 15, no. 3, pp. 243–268, 2005.
15. A. Holzinger and K. Miesenberger, "Mobile computing in healthcare," Universal Access in the Information Society, vol. 9, pp. 1–4, 2010.
16. S. M. Riazul Islam et al., "The Internet of Things for healthcare," IEEE Access, vol. 3, pp. 678–708, 2015.
17. J. Han and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2012.
18. A. Dixit and S. Jain, "Performance analysis of lightweight web services," International Journal of Computer Science, vol. 9, no. 4, pp. 112–118, 2017.
19. G. Coulouris, J. Dollimore, and T. Kindberg, Distributed Systems: Concepts and Design, Addison-Wesley, 2012.
20. P. Kumar and U. Lee, "Security issues in healthcare applications," International Journal of Security and Networks, vol. 7, no. 3, pp. 160–170, 2012.