International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

MediConnect – AI Application "Empowering Health with Intelligence"

Pratiksha P. Jadhav
Information technology
MET's Institute of Technology
Polytechnic, Bhujbal Knowledge City
Nashik, Maharashtra ,India
pratikshajadhav48@gmail.com

Anushka A. Pawar
Information technology
MET's Institute of Technology
Polytechnic, Bhujbal Knowledge City
Nashik, Maharashtra ,India
anushkaap260206@gmail.com

Haniya A. Shaikh
Information Technology
Met's Institute of Technology Polytechnic,
Bhujbal Knowledge City, Adgaon, Nashik,
Maharashtra, India
haniyashaikh433@gmail.com

Mr. V. A. Yeole
Project Guide
Dept. of Information Technology
Met's Institute of Technology Polytechnic,
Bhujbal Knowledge City, Adgaon, Nashik,
India
vijayendray_iot@bkc.met.edu

Abstract—

Medi Connect with AI is an Android-based healthcare application designed to simplify and enhance access to medical services. The app offers key features such as lab test booking, doctor appointment scheduling, health article reading, online medicine purchasing, and AI- powered disease information. By integrating Artificial Intelligence, the app provides users with smart health suggestions and personalized insights. The main objective is to make healthcare more accessible, efficient, and user-friendly, especially for individuals in remote or busy areas.

1. INTRODUCTION

n today's rapidly evolving world, access to healthcare services is crucial, but many individuals face challenges such as long waiting times, high costs, and limited access to medical professionals. To address these issues, *Medi Connect with AI* is an innovative Android application designed to simplify healthcare access. By integrating essential services such as doctor appointment scheduling, lab test bookings, health articles, online medicine purchasing, and AI-powered disease information, the app offers a comprehensive solution to make healthcare more accessible and efficient. The incorporation of artificial intelligence (AI) enhances the user experience by providing personalized health insights, enabling users to identify potential health conditions based on symptoms and seek medical attention more effectively. With a user-friendly

2. REVIEW OF LITERATURE

The integration of artificial intelligence (AI) in healthcare has seen significant growth over the last decade, with AI technologies improving diagnostic accuracy, patient engagement, and operational efficiency. A growing body of research highlights the potential of AI in addressing healthcare challenges such as delayed diagnoses, access to healthcare professionals, and the increasing burden of managing chronic diseases.

enhance service delivery and patient experience. In [1], a mobile health application was developed to assist users with chronic disease management through real-time tracking and AI-based recommendations. The study showed improved health outcomes and user engagement.

In [2], researchers highlighted the importance of telemedicine and AI in diagnosing common diseases, especially during the COVID-19 pandemic. The combination of symptom checkers and AI algorithms proved to be effective in providing preliminary health assessments.

Another study [3] focused on the development of an e-pharmacy app that allowed users to order medicines and consult pharmacists online. It emphasized the growing demand for digital health platforms that are easy to access and use.

3. METHEDOLOGY

The development of *Medi Connect with AI* followed the Agile software development methodology, enabling flexible and iterative progress through continuous feedback and testing. Initially, the requirements were gathered through surveys and analysis of existing healthcare apps. This helped in identifying key features such as lab test booking, doctor appointment scheduling, online medicine ordering, health article access, and AI- powered disease information.

1. **Requirement Analysis**
   - User Research: Conducted surveys and interviews with patients, healthcare providers, and other stakeholders to understand their needs and expectations from a healthcare app. The findings helped define the key functionalities.
   - Feature Definition: Based on feedback, the core features such as doctor appointment scheduling, lab test bookings, health articles, online medicine ordering, and AI-powered symptom analysis were identified.
   - Feasibility Study: Assessed the technical feasibility of integrating AI, real-time booking systems, and secure databases, ensuring the app would meet user needs.
2. **System Design**
   - Architecture Design: Adopted a modular architecture to ensure scalability and maintainability. The app was divided into separate modules like authentication, booking, AI, and pharmacy services for easier maintenance.
   - UI/UX Design: Developed wireframes and prototypes to define the app's flow and visual layout. Focused on making the interface simple, intuitive, and user-friendly.
   - Database Design: Used Firebase for real-time data storage and cloud functionality. SQLite was chosen for offline data storage, ensuring the app remains functional even without internet access.
3. **Technology Stack**
   - Frontend Development: The app was developed using Kotlin, ensuring modern, efficient, and maintainable code for Android devices. XML layouts were used for defining user interface components.
   - Backend Development: Used Firebase for real-time synchronization, user authentication, and cloud storage. Firebase's cloud features allowed for secure and scalable user data management.
   - AI Integration: Implemented TensorFlow Lite, a lightweight machine learning framework, for running AI models directly on the device to provide fast symptom checks and disease predictions.
   - APIs: Integrated third-party APIs for accessing medical content, doctor availability, lab test booking, and online medicine purchasing. These APIs allowed for seamless interaction with external services.
4. **Development Process**
   - Agile Methodology: The development process followed an Agile approach, working in short sprints to focus on delivering specific functionalities. Regular feedback was collected from stakeholders after each sprint to refine the features.
   - Module-wise Development: The app was developed by breaking it down into smaller modules: User authentication, doctor appointment scheduling, lab test bookings, health articles, and AI symptom checkers.
   - Testing and Debugging: Each module underwent unit testing to ensure correctness, followed by integration testing to check how well all components worked together.
5. **AI Module Implementation**
   - Dataset Collection: Gathered datasets from open-source medical resources and research papers to train machine learning models. These datasets included information on common symptoms and related diseases.
   - AI Model Training: Trained machine learning models, including decision trees and logistic regression, to predict diseases based on user input symptoms. The models were optimized for mobile devices using TensorFlow Lite.
   - Natural Language Processing (NLP): Integrated an NLP component that enables the app to understand free-text symptom descriptions from users, converting them into structured data for disease prediction.
6. **Testing and Validation**
   - Unit Testing: Performed unit testing on individual modules (e.g., user authentication, symptom checker, etc.) to ensure each component worked as expected.
   - Integration Testing: Conducted integration testing to ensure that the app's various

components (authentication, booking system, AI model) interacted correctly without errors.

- User Testing: Involved real users to test the app's usability, ensuring it was easy to navigate, responsive, and met their healthcare needs. Feedback from testing helped refine the app's design and functionality.
- Security Testing: Focused on testing the app's data privacy and security measures, ensuring user data (especially health information) was securely stored and encrypted.

7. Deployment

- Beta Testing: Released a beta version of the app to a select group of users for real-world testing. This phase allowed for identifying and fixing bugs, improving performance, and refining features.
- Final Deployment: After ensuring the app's stability and functionality, the final version was prepared for release on the Google Play Store. Continuous monitoring was planned for tracking performance and user feedback post-release.

4. SOFTWARE TESTING & DEPLOYMENT

- **Software Testing**

Testing is a critical phase in the software development lifecycle, ensuring that the application works as expected and meets user requirements. For *Medi Connect with AI*, a variety of testing methods were employed to ensure the app's functionality, usability, security, and performance.

a. Types of Testing Performed

- Unit Testing: Unit testing was conducted on individual components and functions to verify that each part of the app works as intended. For example, testing the logic of symptom input processing and AI-based disease prediction algorithms ensured that they produced the correct outputs.
- Integration Testing: After the individual components were tested, integration testing was performed to ensure that the modules functioned correctly when combined. This included testing the interaction between the user authentication system, doctor appointment booking system, AI module, and external APIs used for health articles and medicine purchases.
- Functional Testing: This form of testing verified that the app's features meet the functional requirements. It involved testing core functions such as login/logout, booking doctor appointments, purchasing medicines, and accessing health articles.
- UI/UX Testing: Given the importance of user experience, UI/UX testing was carried out to ensure that the design was intuitive and accessible. This testing focused on layout consistency, navigation flow, and overall ease of use. Feedback from real users was gathered during this phase to identify potential areas of improvement.
- Security Testing: As the app deals with sensitive user data, comprehensive security testing was conducted. This included:
  - Penetration testing to identify vulnerabilities in the app's code and communication channels.
  - Data encryption testing to ensure that user data is securely stored and transmitted.
  - Authentication and Authorization testing to verify that only authorized users can access sensitive data and services.
- Performance Testing: Performance testing was performed to ensure that the app can handle multiple users and large amounts of data without crashing or slowing down. This included:
  - Load testing: To test how the app handles heavy traffic and multiple simultaneous requests.
  - Stress testing: To determine the app's ability to recover from failure situations, such as when servers are overloaded or the network connection is lost.
- User Acceptance Testing (UAT): User acceptance testing was conducted by a group of real users who tested the app in real-world scenarios. The goal of UAT was to ensure that the app meets user expectations and functions as intended. Feedback from UAT helped identify minor usability issues and confirm the app's overall quality.

b. Bug Tracking and Issue Resolution

During the testing phase, a bug-tracking system was used to document, prioritize, and manage bugs and issues identified by testers. Issues were categorized based on severity, and the development team worked to resolve them promptly. After each fix, the app was re-tested to ensure that the issues were resolved without introducing new problems.

- **Deployment**

Once testing was successfully completed, and the application was deemed stable, the next step was to deploy *Medi Connect with AI* for public use. The deployment process was broken down into several phases:
There are toral 3 phases is follows as :

a. Beta Testing

Before the official release, the app underwent a beta testing phase. The beta version of the app was made available to a small group of users to identify any remaining bugs, gather feedback on usability, and ensure that the app functioned correctly in real-world conditions. The beta testing helped identify potential performance bottlenecks and other issues that weren't discovered in earlier testing stages.

b. App Store Submission

After addressing issues found during beta testing, the app was ready for official release. The app was submitted to the Google Play Store, which involved:

- App Store Optimization (ASO): Creating an appealing app description, screenshots, and promotional materials to attract users.
- Compliance Check: Ensuring that the app complies with all Google Play Store guidelines, such as user data privacy, content policies, and security requirements.
- APK Upload: Uploading the Android Package (APK) file to the Google Play Store, along with necessary metadata like version information and privacy policy.

c. Continuous Monitoring and Maintenance

Once deployed, continuous monitoring of the app's performance was essential to ensure a seamless user experience. The development team used tools such as Firebase Analytics and Crashlytics to track user interactions, monitor for crashes or bugs, and analyze performance issues. Based on this data, the team was able to release periodic updates to enhance the app's functionality and fix any bugs that surfaced post-launch. Maintenance tasks include:

- Bug Fixes: Addressing any issues reported by users after deployment.
- Feature Updates: Rolling out new features and improvements based on user feedback and emerging trends in healthcare.
- Security Patches: Regular updates to ensure the app remains secure and compliant with the latest privacy regulations.

d. Post-Deployment User Feedback

After the app was published on the Google Play Store, the development team monitored user reviews and ratings to gather valuable feedback. This feedback helped in prioritizing future updates and improvements. The team also offered customer support through in-app contact forms and support channels to assist users with any issues or inquiries.

5. RESULT & DISCUSSION

The development and deployment of *Medi Connect with AI* yielded successful results in terms of functionality, user satisfaction, and system performance. The application effectively integrated multiple healthcare services—such as doctor appointment booking, lab test scheduling, AI- powered symptom analysis, online medicine purchasing, and access to health-related articles—into a single, user-friendly Android platform.

6. CONCLUSION

The development of *MediConnect with AI* represents a significant step toward transforming the traditional healthcare system by integrating artificial intelligence with mobile technology. This application effectively simplifies access to healthcare services such as lab testing, doctor consultations, medicine purchases, and health education, all within a unified digital platform. The incorporation of AI-driven features like symptom checking, personalized recommendations, and virtual assistants enhances user experience, improves diagnostic accuracy, and promotes timely interventions.

Through real-time health insights and seamless connectivity between patients and healthcare providers, MediConnect addresses existing gaps in healthcare accessibility, especially in remote and underserved areas. Furthermore, by prioritizing data privacy, secure cloud integration, and compliance with healthcare regulations, the app ensures both reliability and trustworthiness.

In conclusion, MediConnect with AI not only demonstrates the potential of AI in healthcare but also sets a foundation for future innovations in digital health. Ongoing enhancements, including the integration of wearable data, voice-enabled diagnostics, and advanced predictive analytics, will further strengthen its role in delivering intelligent, accessible, and proactive healthcare solutions.

7. REFERENCES

1. J. Jiang, X. Cao, and Y. Huang, "An Intelligent Mobile Health Monitoring System Using Machine Learning," *IEEE Access*, vol. 7, pp. 136552–136563, Sep. 2019, doi: 10.1109/ACCESS.2019.2942551.
2. D. G. Lowe et al., "Artificial Intelligence in Healthcare: Past, Present and Future," *Journal of Medical Systems*, vol. 43, no. 3, pp. 1–14, 2019, doi: 10.1007/s10916-019-1233-7.
3. A. Esteva et al., "A guide to deep learning in healthcare," *Nature Medicine*, vol. 25, pp. 24–29, Jan. 2019, doi: 10.1038/s41591-018-0316-z.
4. A. Vellido, "Societal Issues Concerning the Application of Artificial Intelligence in Medicine," *Kidney Diseases*, vol. 5, no. 1, pp. 11–17, 2019, doi: 10.1159/000492882.
5. S. A. Kumar, R. V. Dharani, and R. S. Anand, "Security and Privacy in Healthcare Applications using Blockchain Technology," *IEEE Access*, vol. 9, pp. 19090–19103, 2021, doi: 10.1109/ACCESS.2021.3055144.
6. A. Rajkomar, J. Dean, and I. Kohane, "Machine Learning in Medicine," *New England Journal of Medicine*, vol. 380, pp. 1347–1358, Apr. 2019, doi: 10.1056/NEJMra1814259.
7. S. Dash, S. K. Shakyawar, M. Sharma, and S. Kaushik, "Big Data in Healthcare: Management, Analysis and Future Prospects," *Journal of Big Data*, vol. 6, no. 1, pp. 1–25, 2019, doi: 10.1186/s40537-019-0217-0.
8. Health Level Seven International (HL7), "HL7 FHIR: Fast Healthcare Interoperability Resources," 2022. [Online]. Available: https://www.hl7.org/fhir/
9. World Health Organization (WHO), "Ethics and Governance of Artificial Intelligence for Health," Geneva, Switzerland, 2021. [Online]. Available: https://www.who.int/publications/i/item/9789240029 200
10. Google Health AI, "Developments in AI for Symptom Checking and Disease Prediction," Google Health, 2022. [Online]. Available: https://health.google/technologies/ai/