APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 6 No. 1 (2025) 2416-2426
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

No-Code Mobile Chatbot App Development Using MIT App Inventor

Noraisyah Abdl Aziz[1*], Rashidah Mokhtar[1], Nooradilla Abu Hasan[2], Muhammad Nur Iman Muslim Reymie[1]

[1] *Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Cawangan Johor, Segamat, 85000 MALAYSIA*
[2] *Faculty of Computer and Mathematical Sciences Universiti Teknologi MARA Cawangan Negeri Sembilan, Seremban, 70300, MALAYSIA*

*Corresponding Author: norai477@uitm.edu.my
DOI: https://doi.org/10.30880/aitcs.2025.06.01.136

Article Info

Received: 3 June 2025
Accepted: 16 June 2025
Available online: 30 June 2025

Keywords

Chatbot app, MIT App Inventor, Block programming, Mobile app development

Abstract

Nowadays, Artificial Intelligence (AI) is making a significant impact on the advanced technology landscape in education, including personalized learning, intelligent tutoring systems, automated grading, predictive analytics, and chatbots. This paper presents the design and development of a chatbot mobile app that eases the search for information using text-based and voice-enabled information retrieval. The development process, adapted from the Mobile application development lifecycle, encompasses key phases such as identification, design, development, prototyping, testing, deployment, and maintenance. MIT App Inventor is used as a development tool, offering a visual environment with minimal coding experience required. The app demonstrates simplified AI integration in mobile solutions. This study offers insights gleaned from overcoming challenges in integrating diverse tools and AI functionalities, aiming to inspire young developers to make an effort to develop practical AI applications.

1. Introduction

The term "chatbot" is a linguistic blend of "chat" and "robot", initially referred to as a text-based dialogue system designed to mimic human conversation [1]. These early versions, primarily computer programs, utilized input and output masks to create a mobile user experience that mimicked a real-time conversation. However, chatbots have significantly evolved beyond this basic text-based interaction. Modern research delves deeper into their inner workings, exploring aspects such as artificial intelligence (AI), natural language processing (NLP), and machine learning (ML). This multifaceted approach enables chatbots not only to respond but also to learn and adapt over time, resulting in more personalized and engaging interactions [2].

In recent years, chatbots have emerged as powerful tools for enhancing user engagement and automating services across various industries, including customer support, education, and healthcare. However, developing functional chatbot applications traditionally requires programming skills and technical expertise that can be a barrier for non-technical users. With the advent of no-code development platforms, such as MIT App Inventor,

there is an opportunity to democratize chatbot development by enabling users with minimal coding experience to design and deploy mobile chatbot applications.

MIT App Inventor is a visual programming environment that allows users to create mobile applications using a drag-and-drop interface. Despite its growing popularity among educators and novice developers, the potential of MIT App Inventor to build sophisticated chatbot applications has not been thoroughly explored in the academic literature. This paper aims to address this gap by developing a no-code mobile chatbot application using MIT App Inventor and evaluating its capabilities.

The primary purpose of this study is to demonstrate the feasibility of creating a chatbot app with MIT App Inventor for searching information using text and voice features. By demonstrating the feasibility of building chatbots without traditional programming, this study contributes to the broader field of accessible technology development, offering insights for educators, developers, and small businesses seeking to leverage chatbots for their specific needs.

The structure of this paper is as follows: Section 2 reviews prior studies, Section 3 details the development process using MIT App Inventor, Section 4 explores the features of the Chatbot app, and Section 5 discusses the limitations of the study, concluding with key insights and future directions.

2. Related Work

In recent years, the demand for interactive and user-friendly mobile applications has become increasingly prevalent. Chatbots can be designed to accept both voice and text commands, providing users with flexibility in how they interact with the application, and has positioned it as modern digital communication. Indeed, chatbots have become ubiquitous virtual assistants that can engage in natural language conversations to address user queries and provide valuable information [3]. The voice command is handy for users who prefer hands-free interaction or have difficulty typing, and can enhance the overall user experience [4].

The development of a robust chatbot mobile app demands integrating various technologies and methodologies to create an efficient, user-friendly, and responsive communication tool. The process typically involves selecting suitable frameworks, algorithms, and design methodologies to meet specific user needs and operational objectives. MIT App Inventor stands out as an innovative online platform, proving as a powerful tool for developing mobile applications, particularly in the realm of chatbots [5].

The visual programming approach of MIT App Inventor, where users can create applications by dragging and dropping components and using a visual block language to program behavior, aligns well with developing a chatbot-enabled mobile app. The development of the chatbot mobile app leverages the capabilities of MIT App Inventor, which has been recognized for its ability to facilitate fast iterative design and its applications in various educational domains, from mathematical applications to multiple-choice examination systems [6]. MIT App Inventor offers a no-code, drag-and-drop interface, making it accessible to beginners and simplifying app development [7]. MIT App Inventor facilitates rapid development and accessibility, making it an ideal tool for both novice and experienced developers.

Specifically, this platform has become increasingly relevant as it supports the integration of artificial intelligence (AI) functionalities, such as those provided by OpenAI APIs, enabling the creation of sophisticated applications like chatbots [8]. These chatbots leverage Natural Language Processing (NLP) to understand and respond to user queries, thereby enhancing both user and application efficiency [4]. Furthermore, the simplicity of MIT App Inventor encourages innovation and experimentation, fostering a learning environment conducive to developing AI-based applications [9].

MIT App Inventor has demonstrated its immense value as an educational platform, providing students at secondary schools with a streamlined introduction to the expansive landscape of computer technology and a frictionless coding experience. It encourages young developers to translate innovative ideas into tangible applications. While MIT App Inventor offers a streamlined approach to developing AI-enhanced mobile applications, it is important to consider the limitations of block-based coding in handling complex AI models. Developers may need to integrate external APIs or libraries to achieve more advanced functionalities, which could require additional programming knowledge beyond the capabilities of App Inventor. Nonetheless, the platform remains a powerful tool for introducing AI concepts and applications to a broad audience, particularly for those beginner level in mobile app development.

This study aims to contribute to the body of knowledge by providing a detailed description of the development process of a chatbot mobile app using MIT App Inventor, highlighting the key challenges, solutions, and best practices encountered throughout the project.

2.1 Comparative Analysis of Chatbot Development Platforms

There are some software platforms and frameworks that help build, train and deploy a chatbot which allow to create conversations like questions, answers and flow. It also allows users to add intelligence, deploy the chatbot to a website, app or messaging platform and lastly monitor and improve the chatbot using analytics and testing. No-code platforms, such as Dialogflow, ManyChat, Tidio and Landbot, offer easy drag and drop interfaces for creating simpler chatbots like FAQ or lead capture. In contrast, developer-focused tools require some programming knowledge and offer more control, customization and integration with other software such as Rasa, Microsoft Bot Framework, OpenAI's ChatGPT API.

MIT App Inventor is a block-based programming tool designed to make app development accessible to beginners, including children, by allowing them to create functional Android applications quickly. It emphasizes educational aspects, fostering computational thinking and generative AI skills through workshops and interactive learning experiences. In contrast, chatbot mobile apps, such as those powered by large language models like ChatGPT, focus on autonomous idea generation and decision-making, often requiring human oversight to refine and implement designs effectively.

Table 1 shows a comparative overview of several popular apps that are used for creating chatbots and apps with a visual, low-code, or no-code interface, similar to how MIT App Inventor works.

Table 1: Comparison of tools for Chatbot And App design

Platform	Suitable features	Strengths	Limitations
MIT App Inventor	Educational apps, basic mobile app	Easy-to-learn, block-based	Android only
Voiceflow	Chatbot-specific development	Great for voice and text bot	Not for general apps
Landbot	Web-based and WhatsApp Chatbot	Support API calls and webhooks	Limited AI capabilities (focused more on flow logic)

Voiceflow's strength lies in its intuitive visual interface, which enables users to create chatbots through a drag-and-drop flowchart-style approach [10], similar to MIT App Inventor's block-based approach. This user-friendly design makes Voiceflow accessible to a wide range of users, from developers to non-technical individuals, enabling them to bring their conversational AI ideas to life [11]. It also allows developers to connect to third party applications using APIs, for example connect to Supabase for database [11]. However, it has a limited free version and requires more configuration for AI-based responses [10].

Landbot [12] offers an intuitive drag-and-drop interface, allowing users to create chatbots without coding knowledge. It enables the deployment of chatbots on web platforms, as well as on WhatsApp, Facebook Messenger, and other messaging channels. Features of this app include a no-code chatbot builder, web-based and WhatsApp chatbots, and a simple interface with a drag-and-drop flow builder. The platform stands out for its ease of use, featuring a highly visual interface that simplifies navigation and setup. It supports complex, multi-step conversational flows, ensuring seamless user interactions. With built-in support for API calls and webhooks, it easily integrates with other systems, enhancing functionality. Additionally, the availability of pre-built templates allows for quick deployment and customization, saving time and effort.

In summary, these three platforms are no-code platforms that simplify app development. MIT App Inventor is best suited for beginners, and its educational use encourages computational thinking through the concept of block-based programming. Voiceflow supports both text and voice chatbots across multiple platforms but with limited general app features. Landbot specializes in building web and messaging chatbots with API support. Each tool serves different needs based on user expectations and application goals.

3. Methodology

3.1 Mobile Application Development Lifecycle

This study adopted the mobile application development lifecycle (MADLC) method proposed by Vithani and Kumar (2014) [13], for the development of the Chatbot application. Figure 1 illustrates the distinct stages of this methodology; the identification or planning, design, development and prototyping, testing, deployment, and maintenance.

The process begins with identification, where the Chatbot's purpose, target audience, and use cases are defined, along with the selection of the platform for deployment, such as websites or messaging apps. Following

this, the design phase involved determining the chatbot type (rule-based, AI-based, or hybrid) and designing the user interface, application database, and overall flow diagram.

In the development stage, backend logic was implemented using visual programming like block in MIT App Inventor environment. The testing phase ensures the Chatbot functions correctly through unit, functional, and usability testing while refining responses based on fundamental interactions. Once a prototype is built in the prototyping stage, it is tested with a limited audience to gather feedback and improve performance. The deployment phase focused on hosting the Chatbot by creating the Android PackageKit (APK) file for distribution. Finally, the maintenance phase is crucial to keep the Chatbot updated with new data, improve its accuracy through model retraining, fix any bugs, and enhance user experience based on ongoing feedback. This structured approach ensures that the Chatbot remains functional, accurate, and efficient in delivering its intended service.

.

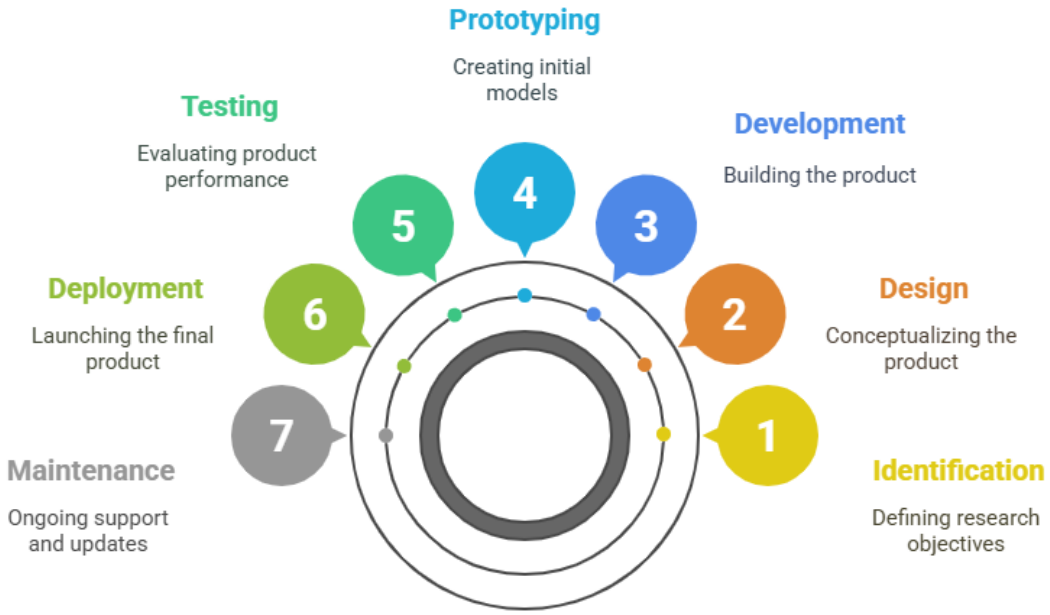

Fig. 1 *Mobile Application Development Lifecycle (MADLC) Model*

Source: (Vithani & Kumar, 2014) [13]

4. Chatbot Mobile Application

4.1 Functional Requirement

The Chatbot app is equipped with five essential features, as outlined in Table 2 below. The first module, Login, allows users to securely access their accounts by providing authentication credentials. It ensures that only authorized users can enter the system, maintaining data privacy and security. Complementing this, the Signup module enables new users to create an account, allowing them to register and gain access to the app's features.

Another key feature is the Voice Assistant, which allows users to interact with the application using voice commands. It enhances accessibility, making it easier for users to navigate and perform tasks hands-free. To complement this, the Voice Assistant Display module ensures that responses from the voice assistant are visually displayed on the screen, providing users with both audio and visual feedback for a more interactive experience. Finally, the Google Screen module allows users to perform searches directly within the app using Google's search engine. This integration eliminates the need for users to switch between applications, ensuring a seamless and efficient search experience. Overall, these functional requirements emphasize user authentication, voice-enabled interactions, and integrated search capabilities, making the application more user-friendly and efficient.

Table 2: *Chatbot app functional requirements*

Module	Description
Login	Allows users to log in to their accounts
Sign up	It provides an option for new users to create an account
Voice assistant	Enables users to interact with the voice assistant for various commands
Voice assistant display	Displays the results of queries made to the voice assistant
Google screen	Allows users to perform searches directly using the Google search engine

4.2 Application Flow and Storyboard

Figure 2 illustrates the storyboard design, which outlines the application flow and interaction within the chatbot app interfaces, including the primary screens and their functionalities. The interfaces are designed with user-friendly features to provide an easy and practical user experience. The sequence begins with the Login Screen, where users can enter their credentials to access the chatbot application. From this point, users are directed to the Menu Screen, which serves as the central navigation hub. This screen provides options for users to interact with the Chatbot through voice commands or conduct Google searches within the app. The Primary Button and Navigation Bar ensure smooth transitions between different functionalities.

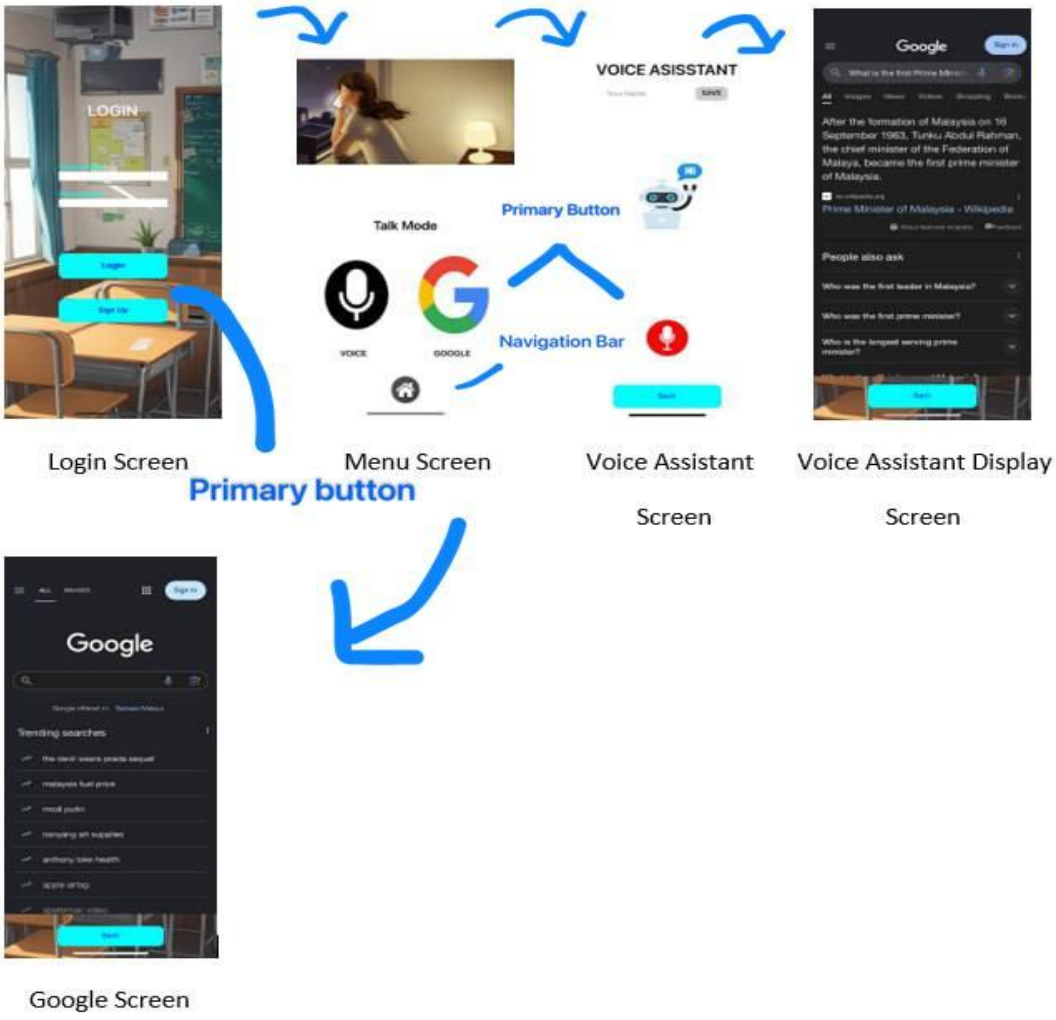

Fig. 2 *Chatbot app storyboard and application flow*

If the user selects the Voice Assistant feature, they are taken to the Voice Assistant Screen, where they can input their name and communicate with the Chatbot through speech. The Chatbot then processes the command and displays relevant information on the Voice Assistant Display Screen, responding to user queries. Alternatively, if the user chooses the Google Search option from the menu, they are directed to the Google Screen, where they can perform web searches within the application. The Primary Button is consistently utilized across different screens, maintaining uniformity and ease of navigation.

Overall, this storyboard effectively outlines the interaction flow within the chatbot application, emphasizing a user-centric design that integrates voice-assisted interactions and direct web searches for an enhanced experience. The structured navigation ensures intuitive usability, allowing users to switch between functionalities seamlessly.

4.3 Interface Design

This section showcases the Chatbot app's interfaces, highlighting its key functionalities. Figure 5 displays the homepage, which serves as both the Login and signup interface for users. The design includes input fields for entering a username and password, along with distinct buttons for logging in or registering a new account.

Figure 6 displays the menu page of the Chatbot, which offers two primary interaction options: voice mode and Google search integration. The interface includes a visually engaging image at the top, enhancing the user experience. The design emphasizes accessibility by incorporating large, easily recognizable icons for voice commands and Google search functionality. These interface elements contribute to the Chatbot's usability by providing intuitive navigation and interaction features, aligning with best practices in user-centered design.

Figure 7 depicts the Google search feature within the Chatbot, enabling users to perform online searches without leaving the application. This functionality enhances user convenience by streamlining access to information directly within the chatbot interface, providing a seamless experience. The inclusion of trending searches further supports a dynamic and interactive experience, ensuring users are informed of current topics.

Concurrently, Figure 8 showcases the Chatbot's voice assistant feature, allowing users to input their names and interact through voice commands. The interface is designed with a minimalist aesthetic, featuring a prominent microphone button for voice input and an animated chatbot illustration, which enhances user engagement. The presence of a "Save" button suggests a personalized experience, potentially enabling the Chatbot to remember user preferences. Together, these features contribute to a user-friendly and efficient chatbot system that leverages both text-based and voice-enabled functionalities to enhance the user experience.

Fig. 5 *Homepage and Login / Signup page of the Chatbot*

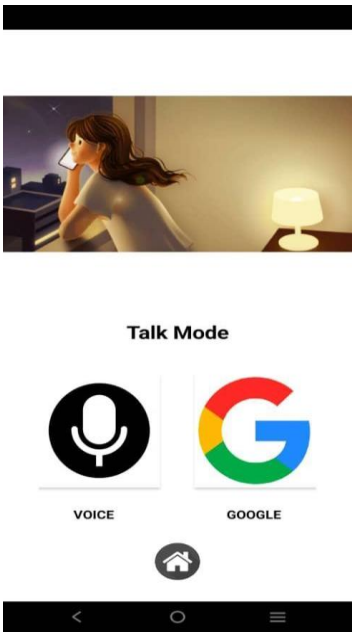

Fig. 6 *Menu page of the Chatbot*

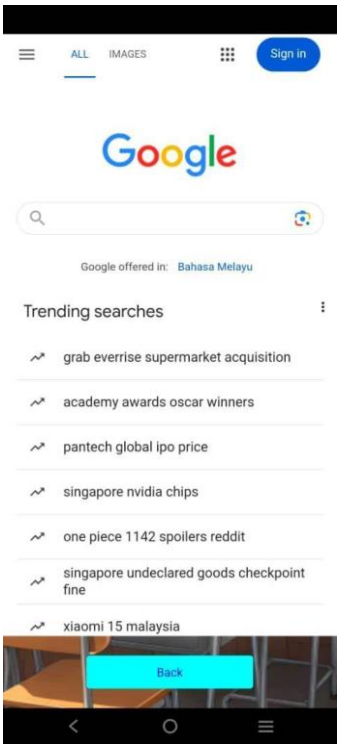

Fig. 7 *Search from the Google page of the Chatbot*

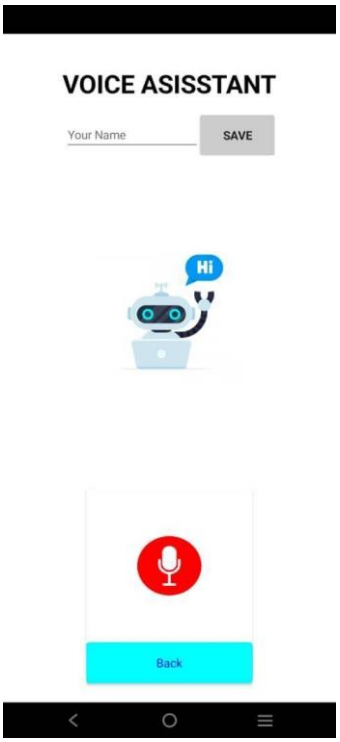

Fig. 8 *Voice function of the Chatbot*

4.4 MIT App Inventor Block Design

The Chatbot app was created using MIT App Inventor, a block-based visual programming environment that allows everyone to build mobile apps with minimal programming skills. It uses the concept of colorful blocks that represent logic structures, events, variables, and functions. Figure 3 illustrates block programming for the login module, where users input their username and password for existing accounts while new users can sign up for a new account. The CloudDB component is used to store and retrieve data online, allowing real-time access to user credentials. By using blocks such as *StoreValue*, *GetValue*, and conditionals (if, compare texts), backend operations like saving new users, validating login credentials, and navigating to a new screen are created.

Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 2416-2426

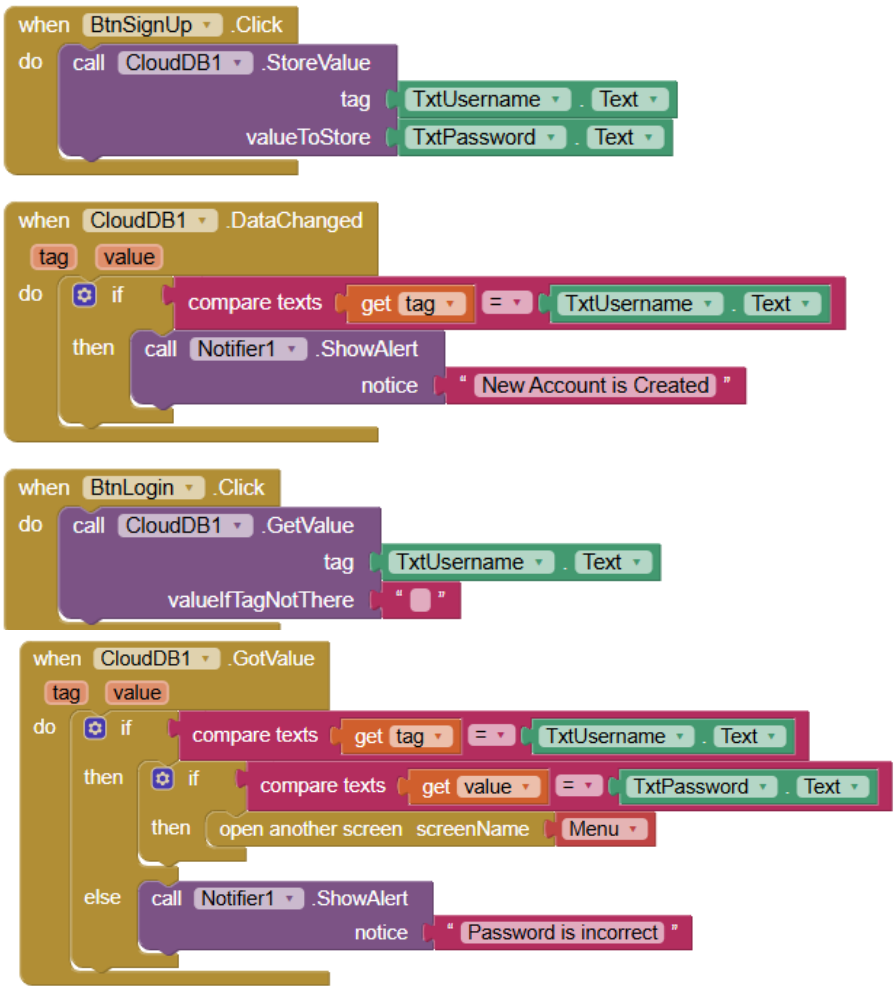

Fig. 3 *MIT App Block - Login Block*

Figure 4 demonstrates a simple block-based program for voice input search function using the *SpeechRecognizer* component. When the user clicks the microphone button, the *SpeechRecognizer1.GetText* function will listen and convert the user's voice into text. Then, it will respond and display output in a text-based format that is set in the block coding in the *TinyDB* database. If the spoken text does not match any predefined phrases, it will navigate to the main menu. The use of colourful MIT App Inventor block simplifies the development process, making it easier to build the chatbot application.

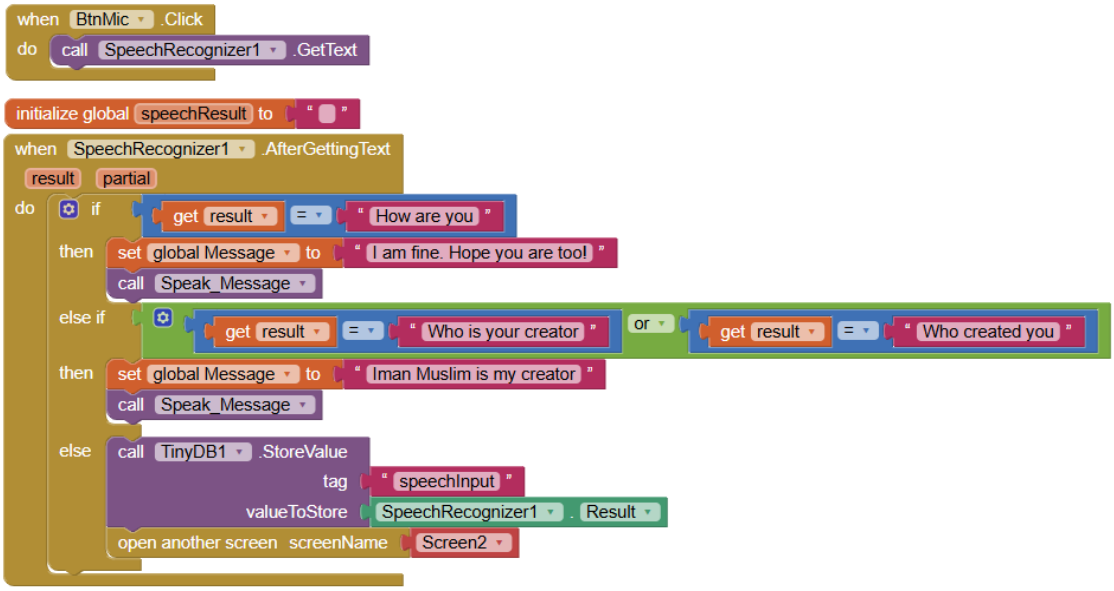

Fig. 4 *MIT App Block – Searching using voice*

4.5 User Acceptance Test (UAT)

This section details the user acceptance test (UAT) results for each module in the chatbot application. Table 3 summarizes the UAT outcomes for the core functionalities of the Chatbot app, encompassing login, signup, voice search, text search, and navigation. All the features in the mobile app were thoroughly tested to ensure they functioned as intended. The error notification was also tested to improve the user experience. This UAT process is essential to ensure the Chatbot is functioning correctly, is user-friendly, and, more importantly, meets the project requirements prior to full deployment

Table 3 *User acceptance test results*

Features	Test case	Expected Results	Actual Result
Login	Enter a valid username and password, click Login	The user is directed to the Menu page	The user is successfully redirected to the Menu page
Login - Invalid Input	Enter an invalid username/password, click Login	Error message "Password is incorrect" is shown	Error message appears as expected
Sign Up	Enter a new username and password, click Sign Up	An account is created, and a confirmation message is shown	Account creation is confirmed with an alert message
Navigation	Navigate from the Login page to the Menu page	The menu page is displayed with voice and Google options	The menu page appears correctly with options
Voice Mode	Tap the microphone icon and say, "How are you?"	Bot replies: “I am fine. Hope you are too!”	Correct voice response is heard
Voice Mode - Unknown	Speak something not predefined	Input is stored, and the user is redirected to Screen2	Unrecognized input was saved and the screen redirected
Google Mode	Tap Google icon	Opens Google or performs keyword search operation	Display results in Google based on keywords

5. Discussion and Conclusion

This study demonstrates the feasibility of developing an AI-embedded chatbot mobile app using the no-code MIT App Inventor platform. The Chatbot effectively delivered both text and voice-based information retrieval features, with the adoption of the Mobile Application Development Life Cycle (MADLC) ensuring a systematic development process. While the chatbot successfully delivered core functionalities, key challenges emerged in optimizing voice recognition accuracy across diverse user pronunciation and management of data flow within the App Inventor environment. Despite the successful development of chatbots, the limitation of this study is that the core functionalities only cover text-based and voice recognition for information searching.

Future work avenues can focus on expanding the chatbot's capabilities, such as image search, using lens to scan objects, sharing search results via the WhatsApp tool, and may include natural language processing (NLP) integration and a suitable API framework for more dynamic features. Concurrently, the chatbot interface needs to be further enhanced to improve user engagement and aesthetic appeal, aligning with the principle of user-centered design.

In conclusion, this study highlighted the potential of MIT App Inventor as a mobile development tool for AI-driven innovation, particularly for beginners and students. This study reaffirms the importance of user-centered design principles for mobile application development to address usability and user experience for sustained engagement.

Acknowledgment

The authors would like to thank the Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Cawangan Johor, for its support.

References

[1] M. Al-Amin, M. S. Ali, A. Salam, A. Khan, A. Ali, A. Ullah, and S. K. Chowdhury, "History of generative Artificial Intelligence (AI) chatbots: past, present, and future development," *arXiv preprint* arXiv:2402.05122, 2024. [Online]. Available: https://arxiv.org/abs/2402.05122

[2] D. Zumstein and S. Hundertmark, "Chatbots—An Interactive Technology for Personalized Communication, Transactions and Services," *IADIS Int. J. WWW/Internet*, vol. 15, pp. 96–109, 2017.

[3] R. Khan and A. Das, "Introduction to chatbots," in *Build Better Chatbots: A Complete Guide to Getting Started with Chatbots*, Berkeley, CA, USA: Apress, 2017, pp. 1–11.

[4] N. Pavitha, P. Bhatele, S. Desai, and H. Pande, "Design and implementation of multipurpose Chatbot," in *Proc. 2022 4th Int. Conf. Smart Syst. Inventive Technol. (ICSSIT)*, Tirunelveli, India, Jan. 2022, pp. 1332–1337, doi: 10.1109/ICSSIT53264.2022.9716306.

[5] E. W. Patton, M. Tissenbaum, and F. Harunani, "MIT App Inventor: Objectives, design, and development," in *Computational Thinking Education*, Singapore: Springer Singapore, 2019, pp. 31–49.

[6] B. M. A. Amer and H. Chouikhi, "Smartphone application using a visual programming language to calculate solar drying parameters," *Sustainability*, vol. 12, no. 19, p. 8148, 2020, doi: 10.3390/su12198148.

[7] M. Kaddipujar, J. Rajan, and B. D. Kumbar, "Mobile application development using MIT App Inventor: An experiment at Raman Research Institute Library," 2022.

[8] J. Gao, C. Su, E. Miller, K. Lu, and Y. Meng, "Rapid Mobile App Development for Generative AI Agents on MIT App Inventor," *arXiv preprint* arXiv:2405.01561, 2024. [Online]. Available: https://arxiv.org/abs/2405.01561

[9] D. Y. J. Kim, A. Zhou, Y. Sudo, and K. Takano, "Advancing Mobile App Development and Generative AI Education through MIT App Inventor," in *Proc. CTE-STEM*, 2024.

[10] Guo et al., “VoiceFlow: Efficient Text-to-Speech with Rectified Flow Matching”, *arXiv preprint* arXiv:2309.05027v1, 2023.

[11] S. Horvat, T. Horvat, L. Havaš, and D. Crčič, “AI-powered chatbots for enhancing accessibility in higher education—Design and implementation,” *Elektrotehniški Vestnik*, vol. 92, no. 1/2, pp. 61–67, 2025.

[12] Landbot, “Landbot: No-code chatbot builder for websites & messaging apps.” [Online]. Available: https://landbot.io/chatbot-platform [Accessed: May 30, 2025].

[13] T. Vithani and A. Kumar, "Modeling the mobile application development lifecycle," in *Proc. Int. MultiConf. Eng. Comput. Sci.*, vol. 1, pp. 596–600, Mar. 2014.