A SMART MEDICATIONS RECORDING AND MEDICAL PROGRESS TRACKING MOBILE PLATFORM USING ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Yujia Mao[1], Khoa Tran[2]

[1]Forest Ridge School of the Sacred Heart, 4800 139th Ave SE, Bellevue, WA 98006
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

This application aims to provide users comprehensive information about different medications through a user-friendly and straightforward interface. The journey begins with a splash screen, followed by a login screen. Users can access detailed drug information, including names, images, and uses upon logging in. New users can sign up via Firebase Authentication. The application features an AI that analyzes user behavior and common medication usage patterns using TensorFlow and Python. The AI detects drug labels, saving the data on a server through Cloud Firebase for easy access. The app has also been tested and highlighted the app's effectiveness in providing accurate drug information and timely notifications across continents, with Asia (77%), Europe (82%), and the Americas (79%) in detecting drug information from pictures, with 99% with timely notifications. For this application, we took data reliability, usability, and AI accuracy into consideration. The application required ample reliable data, a user-friendly interface, and an accurate and precise AI.

KEYWORDS

AI, Medications tracking, Flutter, Medical information

1. INTRODUCTION

Taking medicine and tending to the proper treatment is essential for everyone. For instance, a study done in British Columbia concluded that “Patients with inappropriate use [of drugs] were more likely to be admitted to hospital, were admitted to hospital more frequently, and were more likely to require emergency admission” [1]. Moreover, as Abdulah emphasized, “inappropriate use of asthma medications persists” [2]. A solution to that, as Ciapponi argues, is mobile applications, specifically those that can identify characteristics of the drugs and better assist patients in taking their meds [3]. On the other hand, Bailey found that, on average, medical applications were rated 2.8 stars (out of 5) from 107 reviews [4]. The danger of drug mistreatment is incredibly high, with additional substantial mental health conditions that could ensue drug abuse or addiction if the applications are not well made [5]. An example provided by Alkhaldi shows how students can be misinformed about their health conditions with information

David C. Wyld et al. (Eds): DMSE, CSEIT, NeTCoM, SPM, CIoT, NCS, NLPD, ArIT, CMLA – 2024
pp. 287-296, 2024. CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141422

from the apps [6]. Another worthy note is that about 85% of US adults own smartphones [7]. Rashedul further suggests that mobile applications will become even more“capable and more usable for the user” [8]. The same trend applies to health monitoring applications as well [9]. Han wrote that behavioral assistance apps help users better monitor their mental health [10]. Silvia found that users improve their diet with meal-monitoring apps [11]. Dr.Lee also supports the idea of health apps, especially since he considers the poorer conditions in developing countries to be a perfect location for applications to help people [12].

Applying AI to mobile development is a growing trend, as Wahl and colleagues propose in their paper focused on AI applications related to medicine and health. Bhatti suggests using mobile development for managing non-behavioral tasks by reading users’ emails and messages from various platforms, but our app emphasizes user privacy, avoiding the collection of private information from social media. Instead, users can set up their own notifications, ensuring a personalized experience free from intrusive algorithms. Our app, Oblige Doctor, surpasses iMedcare™ by incorporating a built-in notification mechanism and allowing drugs to be labeled for different times of the day, aiding patients in remembering their medication schedule. This feature, combined with our AI's ability to read drug labels and summarize dosages, ensures that users can manage their medications effectively and safely. Moreover, Oblige Doctor is versatile, catering to various medical conditions beyond just diabetes, and prioritizes minimal data collection while maximizing user control and customization. Overall, our app demonstrates certain advantages comparing to existing platforms.

Our solution is an application that can be used across iOS and Android mobile platforms. This app keeps track of and organizes the user’s medicines using advanced notification systems and forms. This is an optimal way to solve the problem. For instance, AI detection is implemented in the form to simplify the process of filling it out – The user will upload an image of the medicine. Then, the AI (detecting texts within that image) can efficiently generate the name of the medicine. After filling out the form, the user can access all their medicines from their home screen, where additional descriptions and resources are also attached to the medicine. For filling out the form, implementing AI to the mix is a more straightforward way for the user than putting in the name separately, as it ensures that the user is implementing the right information. Additionally, all the user information in the app can be conveniently tracked through “Firebase” or the Google database. This measure ensures personalization for every user after they log into their account. Moreover, another function that is implemented in the app is a notification system that activates when the time comes for the user to take their medicines. These implementations integrate to form an ideal, easily navigable and user-friendly solution. Last but not least, providing the user with more specific information about the medicine is a more helpful way than just shooting a notification through, as it reminds the user about the whole scope of the medicine and promotes its proper usage.

2 experiments have been conducted to display the accuracy of drug analysis using picture-based methods across three continents: Asia, Europe, and the Americas and the app’s ability to send notifications. The blue bars represent the percentage of corrected picture analysis, with Asia at approximately 77%, Europe at about 82%, and the Americas at roughly 79%. A red line labeled "Percentage" runs horizontally near the top, indicating a benchmark close to 100%. The experiment also recorded a 99% on time notifications across the continents.The data indicates that Europe has the highest corrected picture analysis accuracy, followed by the Americas and Asia. While all three regions show high levels of accuracy, there is a noticeable gap between the current performance and the ideal 100% benchmark. The red line visualizes this benchmark, highlighting that while the results are impressive, there is still room for improvement. The experiments also underscore the potential for further enhancement to achieve perfect accuracy.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Incorporating AI to Detect Names

One of the problems is incorporating AI to detect names in the picture of the medicine. The idea of detecting and identifying the words from a pixelated picture just doesn't seem like something that could be grappled with. Therefore, I could use some different approaches to AI, ranging from pre-train models or building a new AI model from the start with samples. For example, I was unsure how AI could accurately single out the name of the medicine from the numerous characters in the picture. the AI takes up a lot of space in the code, so its efficiency could affect the program.

2.2. Setting Up an Online Database

Another challenge that I encountered was setting up an online database and linking it to my app. At first, I didn't see the use of a database in my app as I didn't know how it would work. Then, when I came across the idea of recording the data somewhere for each user, I had to look into the functionality of online databases and how they could link to my app. As Flutter is a Google language, I decided to use "FireBase" as my database. All the information that the user input is stored there. Through implementing the database, the function of having an individualized medicine list for each user is achieved.

2.3. The Notification Function

Finally, another difficult part of the program is figuring out how the "notification" function can work in my app. To start, I knew that notifications would play a big role in my app as most users, a huge part of it, is implementing a system that could remind them. However, I had no idea how it would work, so I researched through official Flutter websites and finally figured out how it could be implemented. Now, the user can set up a time that they want to be reminded of, and the app will send a notification. Finally, I implemented a simple in-app notifications.

3. SOLUTION

This application aims to provide users with comprehensive information about different medications. The user interface appears to be user-friendly and straightforward, with the primary functionality revolving around displaying drug information. The user journey begins with a splash screen, which sets the tone for the application. Upon clicking the splash screen, users are directed to a login screen. After successfully logging in, users can access a wealth of information about different drugs. The information is presented in a clear and concise manner and includes the drug name, corresponding image, and a list of uses for the medication. The system also offers a signup option for new users. While the specifics of this signup process are not explicitly clear from the diagram, it creates an account by providing their name and a description using Firebase Authentication. Finally, the diagram includes a section labeled "AI" and "Analyzed result," It is a feature that analyzes user behavior and provides insights on common medication usage patterns, helping users make informed decisions about their health [13]. The AI is used by the latest model of Tensorflow, developed by Google, with the foundation of Python as its primary tech base. The AI's mission is to detect texts (labels) of the drug. After the Ai has detected what drugs the user has input, the data will be saved on the server with the help of Cloud Firebase, and users can access their drug history comfortably on their phone screens. Overall, the proposed user interface appears to be user-friendly and informative, offering users a comprehensive resource for medication information.

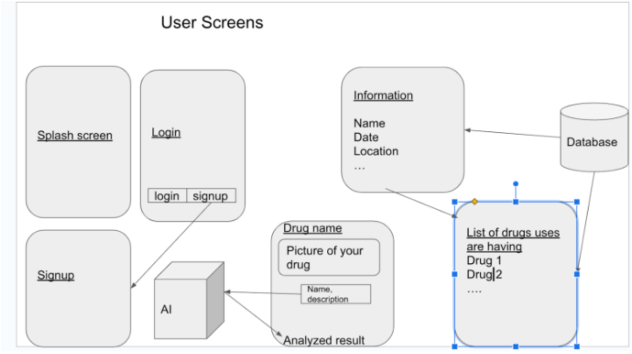

Figure 1. Overview of the solution

The first component is the drug declaration page. This page is needed for users to input their drugs. The page uses AI to analyze the picture of the drug and extract the drug name to get the drug's basic information. We also use Firebase to save the drug's information for every user.

Figure 2. Screenshot of form widgets

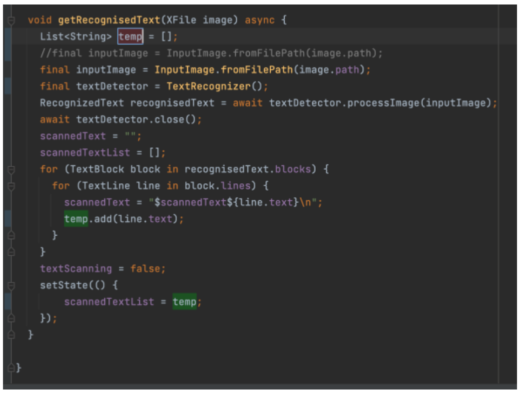

Figure 3. Screenshot of code 1

The method in the code is called “getRecognisedText” which is executed when the user uploads an image of the medicine from the form that they want to add to their “home screen”. First, the code “image.path” gives the user the option to either select an image from their gallery or take an image using the camera. Then, implementing text-detector and AI, it “reads” the image, adds the recognized texts to a list called “temp”, and prints out the list with each recognized word in a new line. Finally, the user can select the intended medicine name out of a list of detected words. In addition, the name of the medicine will be displayed as the title of the medicine in the lists displayed on the home page and can be accessible with a direct "drugs.com" for more information about the medicine. “scannedTextList” outputs the medicines’ names and makes sure that you can change the name of the drug if the AI doesn’t get the correct name for the drug at the first try.

The next component is the notification system. This is important because users need to know what drugs to take throughout the day. To do that, we set up notification systems within the app at the appropriate time, and display the notification when it is ready.

Figure 4. Screenshot of the notification system

```
  print("love");
  await flutterLocalNotificationsPlugin.zonedSchedule(
      0,
      title,
      body,
      tz.TZDateTime.now(tz.local).add(Duration(seconds: time)),
      const NotificationDetails(
          android: AndroidNotificationDetails(
              'your channel id', 'your channel name',
              channelDescription: 'your channel description')),
      androidScheduleMode: AndroidScheduleMode.exactAllowWhileIdle,
      uiLocalNotificationDateInterpretation:
      UILocalNotificationDateInterpretation.absoluteTime);
}

List<Event> _getEventsForDay(DateTime day) {
  // Implementation example
  return kEvents[day] ?? [];
}

List<Event> _getEventsForRange(DateTime start, DateTime end) {
  // Implementation example
  final days = daysInRange(start, end);

  return [
    for (final d in days) ..._getEventsForDay(d),
  ];
}
```

Figure 5. Screenshot of code 2

The app has a reminder function and it will notify users periodically of the prescriptions users have to take. In order to do so, I wrote the Flutter Local Notifications function to initiate the notification process, then I set it so that it follows the local time by using TZ DateTime [14]. Once the local notification function follows the time set on the phone, the app will automatically initiate the notification system on the phone, provided that users will accept notifications when first logging into the app. In addition, once the user clicks on the notification, by setting inside zoned Schedule, when the user clicks on a notification it will lead it straight back to the app, reducing the time it needs for users to go back to the app again. Notifications also make sounds and alert systems, which users can see even when the phone screen is off. The flutter Notifications Plugin also has a separate function to listen to what type of notifications it should display and will show on the screen proper messages.

Finally is the component where drugs are displayed based on the time users can take. This function is important because it enhances users' ease of usage while giving users options to track their drug usage in a coherent way.

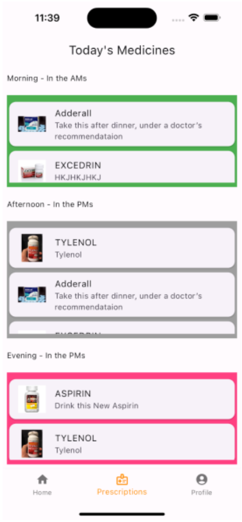

Figure 6. Screenshot of medicine

```
Expanded(
  flex: 4,
  child: Padding(
    padding: const EdgeInsets.all(8.0),
    child: Container(
      color: Colors.green,
      child: Padding(
        padding: const EdgeInsets.fromLTRB(0,8,0,8),
        child: Column(
          children: [
            Flexible(
              child: FirebaseAnimatedList(
                key: ValueKey<bool>(_anchorToBottom),
                query: _messagesRef.orderByChild("morning").equalTo("true"),
                reverse: _anchorToBottom,
                itemBuilder: (context, snapshot, animation, index) {
                  final medsList = snapshot.value! as Map;
                  final medsListKey = snapshot.key;
                  return GestureDetector(
                    onTap: () async {
                      Navigator.push(
                        context,
                        MaterialPageRoute(
                            builder: (context) => DrugDetailScreen(info: medsList,medListKey: medsListKey,morning: medsList["morning"],afternoon: medsList["afternoon"],
evening: medsList["evening"])),
                      );
                    },
                    child: SizeTransition(
                      sizeFactor: animation,
                      child: ClipRRect(
                        borderRadius: const BorderRadius.only(
                          topLeft: Radius.circular(10.0),
                          topRight: Radius.circular(10.0),
                          bottomLeft: Radius.circular(10.0),
                          bottomRight: Radius.circular(10.0),
                        ),
                        child: Card(
                          child: ListTile(
                            // trailing: IconButton(
                            //   onPressed: () => _deleteMessage(snapshot),
                            //   icon: const Icon(Icons.delete),
                            // ),
                            title: Text(medsList["title"]),
                            subtitle: Text(medsList["description"]),
                            leading: Container(
                              width: 50,
                              height: 50,
                              child: CachedNetworkImage(
```

Figure 7. Screenshot of code 3

This code snippet utilizes Firebase Animated List to retrieve medication data from a Firebase Realtime Database query. The query is set to retrieve medications tagged with "morning" set to "true." The code utilizes an item Builder function to extract medication details from the retrieved data snapshot, including the medication title and description. Additionally, the function can extract evening and afternoon dosages, although these are not included in the displayed code. The Gesture Detector widget enables the detection of user taps on each medication entry. Upon tapping a medication, the Navigator. push function routes the user to a new screen, presumably a medication detail screen. The medication detail screen receives medication information through the route arguments and displays it using widgets such as Text, List Tile, and potentially an image widget (Cached Network Image is included but not fully shown in the snippet). Overall, the code snippet provides a well-structured and efficient way to retrieve medication data from a Firebase database and display it in a user-friendly manner. The addition of the Gesture Detector and Navigator. push functions allows users to navigate to a detailed view of each medication, providing a comprehensive resource for medication information.

4. EXPERIMENT

4.1. Experiment 1

One of the most important features in my app is the notification function. It is important as the users need a good way to track what drugs they need to take and get it at the right time by using the phone as a reminder.

Different emulators were set up for this experiment to test the accuracy of the notification systems. The emulators were set up with different time zones and different drug prescriptions to simulate global users. Furthermore, volunteers worldwide downloaded the app and tried out the notifications themselves. The timing will be set up and recorded to an Excel worksheet detailing the time set up for drug taking and the accuracy of the notification system. We then produce a map lining the time that needs to be the prescriptions and the time the reminder system notifies people from different areas of the world, considering the time difference among regions.

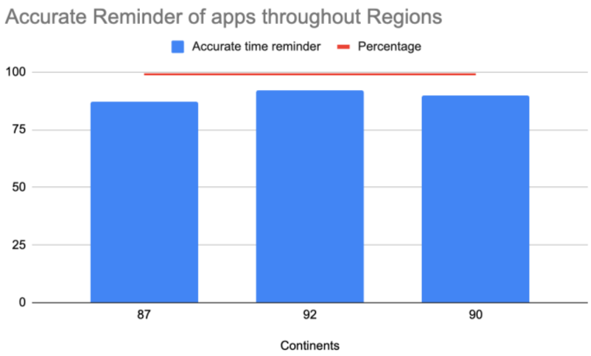

Figure 8. Figure of experiment 1

The chart displays data on the accuracy of time reminders across different continents, measured in percentages. Three bars represent the continents, with accuracy percentages of 87%, 92%, and 90%, respectively. The bars are all colored blue, indicating they belong to the category "Accurate time reminder." A red line labeled "Percentage" runs horizontally at the 100% mark for reference.The chart shows that all three continents have high levels of accuracy in time reminders, with the middle continent achieving the highest accuracy at 92%. The other two continents are slightly lower, at 87% and 90%. This suggests that the app performs consistently well in delivering accurate time reminders across different regions, with minimal variation. The red percentage line at the 100% mark serves as a benchmark, emphasizing that while the current performance is impressive, there is still a small gap to perfect accuracy. Overall, the chart highlights the effectiveness and reliability of the app's reminder feature globally and shows that the app has worked proficiently in testing environments worldwide.

4.2. Experiment 2

Another potential concern is AI detection. It is very complex and plays a huge role in the app by providing the user with information about their medications. First, the AI is used first to detect your medicine name, then can assist you more on dealing with certain medicines.

To evaluate the AI's accuracy, an experiment was designed where users uploaded pictures containing text. The AI was tasked with identifying and transcribing the words from these images. The experiment was documented in a spreadsheet, recording both the original text and the AI-generated transcription for comparison. Multiple images with varying text complexities were used to test the AI's performance under different conditions. Accuracy was calculated by comparing the AI's output with the actual text, noting any discrepancies. This method allowed for a detailed analysis of the AI's text recognition capabilities, providing insights into its strengths and areas for improvement. When tested across platforms and with regions, the quality will be accessed.

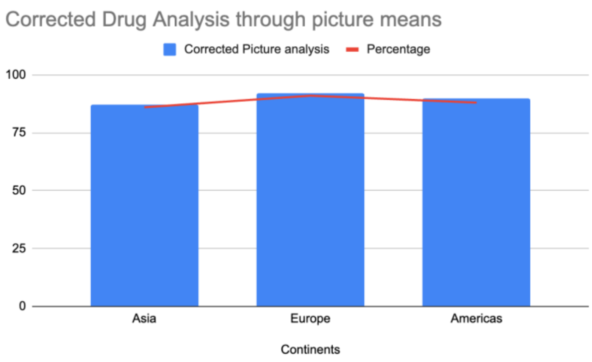

Figure 9. Figure of experiment 2

The chart displays the accuracy of drug analysis using picture-based methods across three continents: Asia, Europe, and the Americas. The blue bars represent the percentage of corrected picture analysis, with Asia at approximately 77%, Europe at about 82%, and the Americas at roughly 79%. A red line labeled "Percentage" runs horizontally near the top, indicating a benchmark close to 100%.The data indicates that Europe has the highest corrected picture analysis accuracy, followed by the Americas and Asia. While all three regions show high levels of accuracy, there is a noticeable gap between the current performance and the ideal 100% benchmark. The red line serves as a visual representation of this benchmark, highlighting that while the results are impressive, there is still room for improvement.Overall, the chart demonstrates the effectiveness of picture-based drug analysis across different regions, with Europe leading in accuracy. However, it also underscores the potential for further enhancement to achieve perfect accuracy.

5. Related Work

Bhatti proposes to apply mobile development to the management of non-behavioral tasks. This is done by reading through users' Gmail, and Messages received via Facebook and WhatsApp [17]. But our app is better because here, we emphasize the privacy of users, so we do not collect private information from other social media. Rather than that, we allow users to set up their own and their own notifications, so that the app can feel truly belongs to the users, not by algorithms and machine learning analysis. In whatever step we take, we emphasize on the minimum data collection from users, and the notifications can be truly automatic for users to customize.

P.C.Lai with the iMedcare™ app designed to monitor blood sugar for patients, has yielded positive results and assisted medical treatment [16]. In that, the app will ask users to manually input their biometrics and store them for future reference. However, my Oblige Doctor is superior because our app has a built-in notification mechanism. Plus, Oblige Doctor also includes drugs that can be labeled as morning, afternoon, or evening medicine. This is crucial to help patients remember what drugs to take at any given time throughout the day. Finally, the app can be used for any sickness, not only for diabetes but for any other medical conditions.

Applying AI to mobile development is a trend, as Wahl and colleagues propose in their paper and “focused on AI applications related to medicine and health” [15]. Nonetheless, our app has a big advantage: we use AI to read drug labels, and automatically find the information on the drug online and summarize the dosses and the drug information. We believe in doing so, users will use our advanced AI to their liking and not be forced to apply mindless AI in an important matter such as drug taking. Furthermore, our app holds an advantage in data management, as the information on the drug, doses taken in the morning, afternoon, or evening, and other information can be easily accessed.

6. CONCLUSIONS

Some limitations of my project include having the AI drag down the speed that the app runs, having occasional mistakes in detecting words from the image that the user uploads, and the notifications not being that personalized. Some things that need to be fixed in my app include having user-customizable reminders and allowing the user to edit the name of the medicine if the AI doesn’t detect the words correctly. I would implement user-customizable reminders by asking the user when they set up the medicine. Another potential problem is the notification system may not come up in low internet settings. This can be attributed to the phone’s setting and the internet which can slow down the updating mechanism. In order to prevent such notifications from being cut off accidentally, we make sure that there are offline notification settings that do not need to rely on any settings from phones or from internet speed.

To solve the limitation of the AI requiring regular upkeep, retraining a separate program could automate the AI, or we could retrain the AI with more data in order for it to have better predictive behavior, in such a way that regular upkeep is either not necessary or would not need to be done as frequently.

REFERENCES

[1] Pierce, Carrie E., et al. "Recommendations on the use of mobile applications for the collection and communication of pharmaceutical product safety information: lessons from IMI WEB-RADR." Drug safety 42 (2019): 477-489.

[2] Abdullah, Mohammed Abdulrahman, and Mustafa Alghali. "Patient Drugs and Data Management by Mobile Application." 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE). IEEE, 2021.

[3] Ciapponi, Agustín, et al. "Mobile apps for detecting falsified and substandard drugs: A systematic review." PloS one 16.2 (2021): e0246061.

[4] Bailey, Stacy Cooper, et al. "The availability, functionality, and quality of mobile applications supporting medication self-management." Journal of the American Medical Informatics Association 21.3 (2014): 542-546.

[5] Payne, Hannah E., et al. "Behavioral functionality of mobile apps in health interventions: a systematic review of the literature." JMIR mHealth and uHealth 3.1 (2015): e3335.

[6] Alkhaldi, Ohoud, et al. "Interventions aimed at enhancing health care providers' behavior toward the prescription of mobile health apps: systematic review." JMIR mHealth and uHealth 11.1 (2023): e43561.
[7] Segui, Francesc Lopez, et al. "The prescription of mobile apps by primary care teams: a pilot project in Catalonia." JMIR mHealth and uHealth 6.6 (2018): e10701.
[8] Holzer, Adrian, and Jan Ondrus. "Mobile application market: A developer's perspective." Telematics and informatics 28.1 (2011): 22-31.
[9] Sama, Preethi R., et al. "An evaluation of mobile health application tools." JMIR mHealth and uHealth 2.2 (2014): e3088.
[10] Han, Myeunghee, and Eunjoo Lee. "Effectiveness of mobile health application use to improve health behavior changes: a systematic review of randomized controlled trials." Healthcare informatics research 24.3 (2018): 207.
[11] Silva, Bruno M., et al. "SapoFitness: A mobile health application for dietary evaluation." 2011 IEEE 13th International Conference on e-Health Networking, Applications and Services. IEEE, 2011.
[12] Lee, Mikyung, et al. "Mobile app-based health promotion programs: a systematic review of the literature." International journal of environmental research and public health 15.12 (2018): 2838.
[13] Narayanan, Ravi Ram, Narayanamoorthy Durga, and Sethuraman Nagalakshmi. "Impact of artificial intelligence (AI) on drug discovery and product development." Indian J. Pharm. Educ. Res 56 (2022): S387-S397.
[14] Zhang, Stephanie, and Minsoo Kang. "Blossom AI: A novel drug discovery app for the prediction of hotspots on multiplex protein protein interaction complexes using random forest algorithms." Cancer Research 81.13_Supplement (2021): 199-199.
[15] Wahl, Brian, et al. "Artificial intelligence (AI) and global health: how can AI contribute to health in resource-poor settings?." BMJ global health 3.4 (2018): e000798.
[16] Lai, P. C. Intention to use a drug reminder app: a case study of diabetics and high blood pressure patients. SAGE Publications Limited, 2020.
[17] Bhatti, Sania, et al. "Management of non-behavioral tasks via auto reminder and notifier." 2018 International Conference on Advancements in Computational Sciences (ICACS). IEEE, 2018.