

# Advanced Development of a Prayer Schedule Bot Application on Telegram Using PHP

Asroni, Cahya Damarjati\* and Sofran Bahrurozi

*Universitas Muhammadiyah Yogyakarta, Jln. Brawijaya, Tamantirto, Kasihan,  
Bantul, Yogyakarta 55183, Indonesia*

*\*Corresponding author: cahyadamarjati@umy.ac.id*

## **Abstract**

*A bot is a web robot or an assistant with artificial intelligence (AI) capabilities. The bots can be used for web assistants, advertisements, and instant messaging. One of the popular instant messaging systems i.e., Telegram, provides a bot feature to enhance the user experience. However, the massive activity in this messaging platform causes users to forget their religious activities, such as praying. Therefore, a prayer schedule bot is needed for reminding users regarding their religious activities. To support the phases of making this praying bot, a system framework is designed using PHP programming language. With some extensions during bot development, the user's location can be collected by the bot, providing a more accurate time presentation. Finally, the bot could be used with the various calculation methods defined by Muslim scholars, providing a preference for users from various countries.*

**Keywords:** *bot; prayer schedule; PHP; calculation method.*

## **1. Introduction**

Social media emerged with various models and uses. In this case, many developers began to create various social media applications with their respective advantages. The use of social media has a significant impact on the world's population; even Indonesia is among the 3rd largest users of social media in the world. According to the Indonesia Dictionary (KBBI), social media is a page or application, allowing users to create and share content or engage in social networks. It facilitates people to communicate at great distances. Technological developments lead to faster and more practical needs, one of which is creating bots that can be done using various tools [1, 2].

Social life involves many activities, ranging from working to support the family and others. People preoccupied with work certainly require proper communication. Those with a busy life in a very dense work field are likely to use social media as a means of long-distance communication [3-5]. One of the most useful social media is Telegram. Telegram is a free and non-profit cloud-based multi-platform instant messaging service. It is one of the many social media available, with the same features as others. However, it provides countless benefits. It can send files of large size and make bots, commonly known as internet robots. The bot is software that can do a command or task automatically via the internet [6].

Apart from many activities at work, Muslims frequently neglect prayer as they merely focus on their social media [7]. Although, humans need rest time during their

intense activities and this prayer time could be used as a reminder to take a rest spiritually. Many developers have created several applications to remind people of prayer, unfortunately, they more prefer to use the Social Media application. Social Media applications, such as Telegram, can create various bots with a remote communication tool at the same time. Hence the authors developed a prayer schedule bot on Telegram entitled “Advanced Development of the Bot Application Prayer Schedule on Telegram Using PHP”.

The remainder of this paper is organized as follows. Section II provides a methodology. In Section III, the results are described. Section IV presents discussions, followed by Section V which summarizes this study.

## 2. Method

### A. Research Stages

The stages designed are useful for carrying out the implementation phase. If, at any stage, there is a problem, a study must be checked and reviewed to examine further whether something is happening to the research results, thereby producing good results.

The initial stage of this research was the establishment of a framework from the beginning to the end. It was followed by determining the location in the world, which was part of data collection. The third phase was to design a system for the bot schedule prayer framework. The next step was to test the message test to monitor the Telegram response. Furthermore, the last stage was the interface design.

### B. Creating a Framework

Before making a bot, the researchers created a framework. Thus, the bot’s thesis and creation were more structured in this workshop series.

### C. System Design

The system design in the development of this Telegram bot employed a flowchart on the Telegram bot workflow. First, before the telegram bot sent a response message to the user, the user would type the command “/ start” which would be sent to the bot. The bot would receive messages from the user and executed it. After the message was executed, the notification appeared from the bot, and a button emerged to run from the features made. Look at Algorithm 1

**Algorithm 1.** Bot Replay Message

---

**Input:** User Message

**Output:** Bot Replay Message

---

```
function setBot (Message)
  if(Message == command bot)
    Switch(Message)
      Case ‘Set Location’
        Bot send current location to the user
        Break
      Case ‘Calculation Method’
        Bot send calculation method selection
        Break
      Case ‘Prayer Schedule’
        Bot send prayer schedule
        Break
      Case ‘Info’
        Bot send information about developer
```

---

---

```
    Break
end if
else
    bot not respond
    return false
end
```

---

### 3. Analysis Results

#### A. Work Program Steps

In the work program steps, the authors have observed various regions that have finally been collected and divided according to continents. The observation was employed to collect regional data. The use of observation was deemed to be more effective due to the large number of regions scattered throughout the world. Therefore, Algorithm 2 is used to set the calculation method preferred by the user from various countries.

#### Algorithm 2. Bot Replay Selection

---

**Input:** Selection of Calculation Method  
**Output:** Bot Replay Selection

---

```
function setMethod (Selection)
    if(Selection == command bot)
        Switch(Selection)
            Case 'University of Islamic Sciences, Karachi'
                Bot set calculation method: 'University of Islamic Sciences, Karachi'
                Break
            Case 'Islamic Society of North America [ISNA]'
                Bot set calculation method: 'Islamic Society of North America [ISNA]'
                Break
            Case 'Muslim World League'
                Bot set calculation method: 'Muslim World League'
                Break
            Case 'Makkah, Umm al-Qura'
                Bot set calculation method: 'Makkah, Umm al-Qura'
                Break
            Case 'Egyptian General Authority of Survey'
                Bot set calculation method: 'Egyptian General Authority of Survey'
                Break
        end if
    else
        bot not respond
        return false
    end
```

---

The results obtained were used to determine the location of users using the schedule prayer bot. The authors sorted and entered location data to determine the time zone. After the required data were collected, a test was performed, in which the response of the Telegram bot would continue to be active or not. The test was successful if the bot could respond and send data according to its duties.

#### B. Database

In this database design, the writers made a table with five columns containing chat\_id, latitude, longitude, time zone, and method. The database aimed to receive an ID from the user and store the data entered.

### C. Testing

The test aimed to check the functionality of the bot. The results of the test were entered into a database. The database structure consisted of five columns, namely chat\_id, lat (latitude), lon (longitude), timezone, and method. In testing, the user tried the bot in the first-order type or clicked “/ start” then the bot would send a message to the database to accept user chat\_id, then clicked the set location button to accept latitude, longitude, and timezone to be entered into the database, and clicked the calculation method button to enter the method selection into the database. The bot created would display four buttons, namely set location, calculation method, prayer schedule, and info, as illustrated in Figure 1.

## 4. Discussion

The performance of the study results was compared with the conventional prayer app to provide a preference for users. Table 1 presents the overall profiles. Based on the table, a prayer scheduler using Telegram bot was preferred for smartphone users who wanted storage efficiency and had an internet connection. As depicted in the table that the first choice had less user data size and communication data.



Figure 1. Bot Interface

The user who prefers the second choice, prayer scheduler conventional app, does not care much about the taken storage and does not have an internet connection. From Table 1, this bot only consumes 50MB of in-app storage and 20 MB of user data. Compared to the conventional application, there is a 66% reduction in user data. This program is more efficient for the user who has a limited storage capacity of their mobile.

Table 1 Performance Comparison

The Apps Compared List	Conventional Prayer Scheduler App	Prayer Scheduler Telegram Bot
Used storage in-app	±50 MB	±50 MB
Used storage in user data	±60 MB	±20 MB
Internet connectivity	Whenever needed	Must available
Communication size	±0.5 MB	±1 KB
GPS feature	available	available
Calculation option	available	available

## 6. Conclusion

This study presents a framework design and bot prototype that serves as a prayer time reminder for Telegram users. The bot is capable of recording the user's location and synchronizing the information in the database. From the experimental results, a simple interface can also be developed to make it easier for users to access location settings, calculation methods, prayer schedules, and developer information. The development of other more complex features such as windows and sound notifications for this chat bots is future works.

## References

- [1] M.-A. Yaghoub-Zadeh-Fard, B. Benatallah, F. Casati, M. C. Barukh, and S. Zamanirad, "User Utterance Acquisition for Training Task-Oriented Bots: A Review of Challenges, Techniques and Opportunities," p. 9
- [2] C. Huda, F. A. Bachtiar, and A. A. Supianto, "Reporting Sleepy Driver into Channel Telegram via Telegram Bot," 2019, pp. 251–256.
- [3] J. C. de Oliveira, D. H. Santos, and M. P. Neto, "Chatting with Arduino platform through Telegram Bot," 2016, pp. 131–132.
- [4] R. R. Buamona, "Auto Response Message Pada Bot Telegram Untuk Pelayanan Sistem Informasi Monitoring Skripsi," Digital Library Teknik Informatika, 2019.
- [5] D. Korotaeva, M. Khlopotov, A. Makarenko, E. Chikshova, N. Startseva, and A. Chemysheva, "Botanicum: a Telegram Bot for Tree Classification," 2018, pp. 88–93.
- [6] S. Perez-Soler, E. Guerra, J. de Lara, and F. Jurado, "The rise of the (modelling) bots: Towards assisted modelling via social networks," in 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE), Urbana, IL, Oct. 2017, pp. 723–728, doi: 10.1109/ASE.2017.8115683.
- [7] E. Abu Bakar, N. J. Draman and A. Z. Saidin, "Value, Religiosity and Behaviour in Social Media," 2018 International Conference on Information and Communication Technology for the Muslim World (ICT4M), 2018, pp. 42-46, doi: 10.1109/ICT4M.2018.00017.
- [8] S. I. Ch'ng, L. S. Yeong, and X.-Y. Ang, "Preliminary Findings of using Chat-bots as a Course FAQ Tool," 2019, pp. 1–5.
- [9] U. Bharti, D. Bajaj, and H. Batra, "Medbot: Conversational Artificial Intelligence Powered Chatbot for Delivering Tele-Health after COVID-19," IEEE Conference Record, p. 6, 2020.
- [10] J. Fahana, R. Umar, and F. Ridho, "Pemanfaatan Telegram Sebagai Notifikasi Serangan untuk Keperluan Forensik Jaringan," Query: Journal of Information Systems, vol. 1, no. 2, 2017.
- [11] S. Kumari, Z. Naikwadi, A. Akole, and P. Darshankar, "Enhancing College Chat Bot Assistant with the Help of Richer Human Computer Interaction and Speech Recognition," p. 7, 2020.
- [12] S. Salvi, V. Geetha, and S. S. Kamath, "Jamura: A Conversational Smart Home Assistant Built on Telegram and Google Dialogflow," 2019, pp. 1564– 1571.