Mobile Usage Guide

Project submitted to the
SRM University - Ap, Andhra Pradesh
for the partial fulfillment of the requirements to award the
degree of

Bachelor of Technology/Master of Technology
In
Computer Science and Engineering
School of Engineering and Sciences

Submitted by
Candidate Name
AP23110010568 — A.Sai Kishore
AP23110010565 — D.Prabath
AP23110010535 — A.Ashwin
AP23110010580 — D.Vamsi.

Under the Guidance of
Dr.KAVITHA RANI KARNENA

SRM University – AP
Neerukonda, Mangalagiri , Guntur
Andhra Pradesh – 522 240
[November,2024]

Certificate

Date: 20-Nov-24

This is to certify that the work present in this Project entitled “ **Mobile Usage Guide** “ has been carried out by **[A.Sai Kishore, D.Prabath, A.Ashwin, D.Vamsi]** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences.**

Supervisor
Dr.Kavitha Rani Karnena

(Signature)
Prof./ Dr.[Kavitha Rani Karnena]

Acknowledgements

Dear madam,

We would like to express my sincere thanks to our beloved madam for her unwavering support and guidance throughout this journey. Your expertise, encouragement, and insightful feedback have been invaluable, shaping not only the project's success but also our growth as individuals and as a team.

This project encapsulates more than just lines of code, it represents a commitment to quality and a pursuit of efficiency. The Mobile Usage Guide, with its modular structure and thoughtful implementation, stands as a testament to the importance of code organization, clarity, and user-friendliness.

To my classmates, I extend my heartfelt thanks for their collaboration and the valuable insights they have shared. Their dedication and enthusiasm have made a significant impact on my learning experience.

Table of contents:

Abstract:

The Mobile Usage Guide program is a user-friendly command-line application designed to provide step-by-step guidance on performing various common tasks on a mobile device. The program offers detailed instructions for activities such as accessing contacts, playing music, sending messages, setting alarms, and more. This guide serves as a practical tool for users who need quick reference or assistance in navigating mobile features efficiently. By offering structured and accessible information, the application aims to enhance user interaction with mobile devices, making technology usage more intuitive.

Introduction:

In today’s digital age, mobile devices have become essential tools for daily communication, productivity, and entertainment. However, due to the extensive features available on modern smartphones, users may sometimes find it challenging to locate or perform specific tasks. The Mobile Usage Guide program is developed to address this issue by providing a straightforward, interactive menu-driven interface that users can access to receive comprehensive guidance on various mobile functions. The program is designed to cater to both novice and experienced users who seek a reliable source for performing essential mobile operations.

- **Interactive Menu Interface**: Provides a simple, easy-to-navigate menu with multiple options for quick access.
- **Comprehensive Guidance**: Offers clear, step-by-step instructions for 13 common mobile tasks.
- **User Input Validation**: Ensures that users input valid choices, improving the program's reliability.
- **Modular Functions**: Each task is handled by a dedicated function, making the code maintainable and scalable.
- **Feedback Mechanism**: Prompts users to decide if they need further assistance or wish to exit the guide.

Methodology:

- The Mobile Usage Guide is developed as a console-based application using C++.
- The program displays a menu with 13 options, each corresponding to a common mobile task.
- Selecting an option triggers a specific function that outputs step-by-step instructions for completing that task.
- Functions are modular and individually handle different actions such as making calls, browsing the internet, setting alarms, etc.
- Input validation is implemented to ensure users provide valid choices.
- A loop structure allows users to return to the main menu for further guidance or exit the application as desired.
- A user feedback mechanism prompts users to choose if they need further assistance before navigating back to the main menu.

Objectives:

The primary objectives of the Mobile Usage Guide program are:

1. **To provide comprehensive assistance** for users navigating key features of their mobile devices, thus simplifying the user experience.
2. **To ensure ease of use and accessibility** through an intuitive, menu-driven interface that supports repeated interactions.
3. **To deliver clear and concise instructions** for each mobile function covered, minimizing confusion and enhancing user confidence.
4. **To serve as a helpful learning tool** for individuals less familiar with smartphone operations, bridging the gap between technology and user proficiency.

By achieving these objectives, the program promotes a better understanding and usage of mobile device capabilities, contributing to improved digital literacy and user satisfaction.

CODE:

```
#include <bits/stdc++.h>


using namespace std;

void displayMenu() {

   cout << "\n\t\t\t\t\tKeypad Mobile Usage
Guide\n\t\t\t\t------------------------------"

       << "\n1. Contacts\n2. Play Music\n3. Open Camera\n4. Send Message\n5.
Browse Internet\n6. Set Alarm"

       << "\n7. Check Battery\n8. Make a Call\n9. Send Email\n10. Check
Weather\n11. Take a Screenshot"

       << "\n12. Use Voice Assistant\n13.
Exit\n----------------------------------\nSelect an option for guidance: ";

}

int getUserChoice() {

   int choice;

   while (true) {

      cin >> choice;

      if (cin.fail() || choice < 1 || choice > 13) {

         cout << "Invalid choice! Please enter a valid option (1-13): ";

         cin.clear(); cin.ignore(numeric_limits<streamsize>::max(), '\n');

      } else return choice;

   }
```

```
}

void contacts() {
   cout << "\nHow to Access Contacts :"
       << "\n- Open the app and navigate to the 'Contacts' tab."
       << "\n- Tap on 'Add New Contact' to create a new contact."
       << "\n- To search for a contact, use the search bar at the top of the screen."
       << "\n- Tap on a contact to view details or call them directly." << endl;
}

void playMusic() {
   cout << "\nHow to Play Music :"
       << "\n- Open the app and go to the 'Music' tab."
       << "\n- Browse through your music library or search for a specific song."
       << "\n- Tap on a song to play it. You can pause, play, or skip using the on-screen buttons."
       << "\n- To play music in the background, simply press the 'Home' button." << endl;
}

void openCamera() {
   cout << "\nHow to Open the Camera :"
```

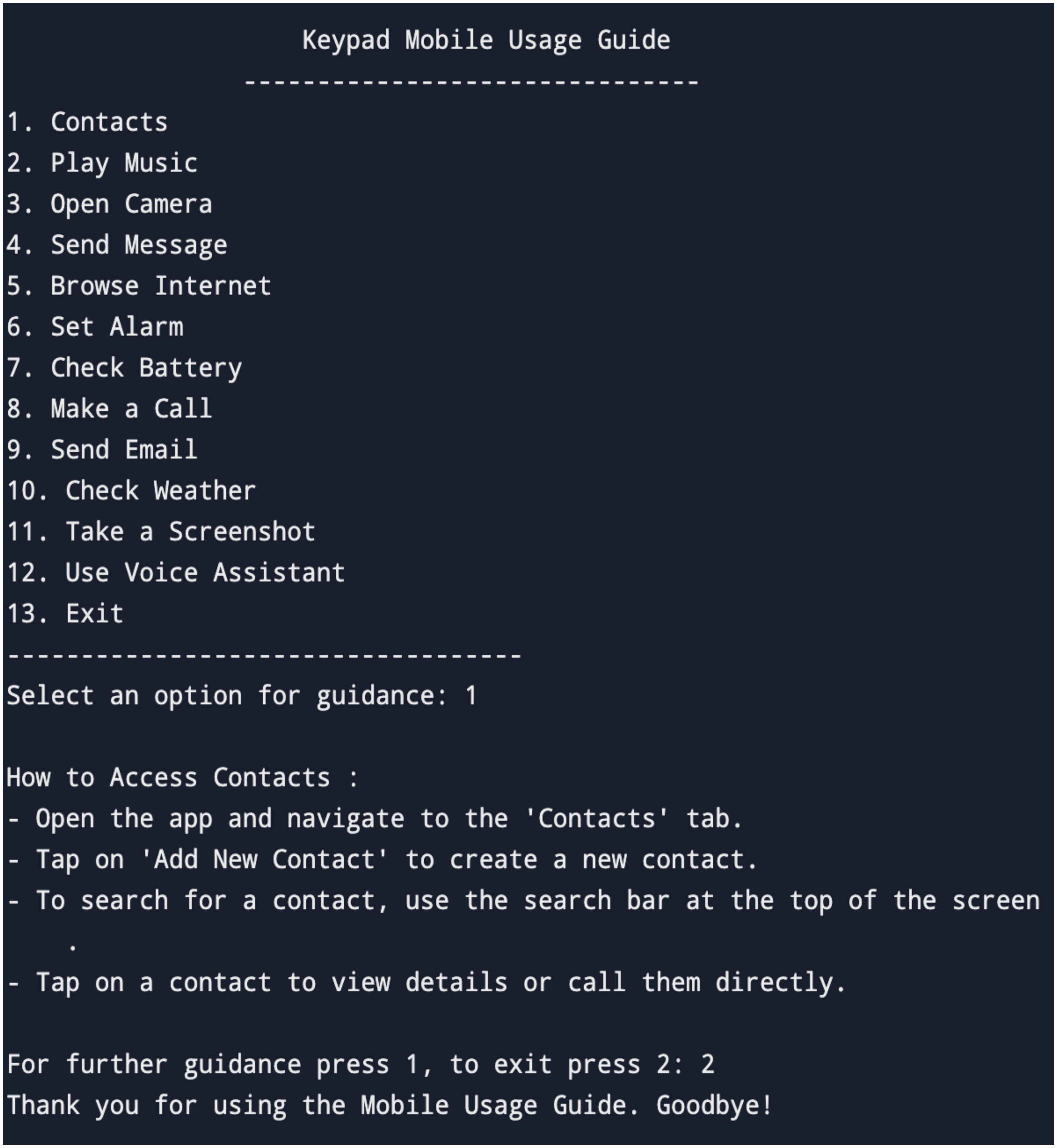
```
        << "\n- Open the app and navigate to the 'Camera' tab."
        << "\n- You can switch between front and back camera by tapping the camera icon."
        << "\n- Tap the shutter button to take a photo or press and hold for video."
        << "\n- Photos will be saved in the 'Gallery' section of the app." << endl;
}
void sendMessage() {
    cout << "\nHow to Send a Message :"
        << "\n- Open the messaging app and navigate to the 'Chats' tab."
        << "\n- Tap the 'New Message' button and select a contact to message."
        << "\n- Type your message in the text box and hit 'Send'."
        << "\n- You can also send images or voice messages by using the icons next to the text box." << endl;
}

void browseInternet() {
    cout << "\nHow to Browse the Internet :"
        << "\n- Open the browser app on your mobile device."
        << "\n- Type the URL or keyword you want to search in the address bar."
        << "\n- You can swipe left or right to navigate through different tabs."
        << "\n- To go back, tap the back button or swipe right from the left edge of the screen." << endl;
}
```

```
void setAlarm() {

    cout << "\nHow to Set an Alarm :"

        << "\n- Open the Clock app and navigate to the 'Alarm' tab."

        << "\n- Tap the '+' button to add a new alarm."

        << "\n- Set the time and choose your desired alarm sound."

        << "\n- Tap 'Save' to set the alarm."

        << "\n- You can edit or delete the alarm at any time by tapping on it." << endl;

}

void checkBattery() {

    cout << "\nHow to Check Battery Status :"

        << "\n- Swipe down from the top of the screen to open the notification shade."

        << "\n- Your battery percentage will be displayed at the top."

        << "\n- To view more details, tap the battery icon and check for battery usage statistics." << endl;

}


void makeCall() {

    cout << "\nHow to Make a Call :"

        << "\n- Open the Phone app and tap on the 'Dialer' tab."

        << "\n- Type the phone number you want to call, or select a contact from the
```

```
 'Contacts' tab."
        << "\n- Tap the green call button to initiate the call."
        << "\n- To end the call, tap the red 'End Call' button." << endl;
}


void sendEmail() {
    cout << "\nHow to Send an Email :"
        << "\n- Open your email app and tap on the 'Compose' button."
        << "\n- Enter the recipient's email address, subject, and the body of your message."
        << "\n- Tap 'Send' to send the email." << endl;
}
void checkWeather() {
    cout << "\nHow to Check the Weather:"
        << "\n- Open the Weather app on your device."
        << "\n- The current weather will be displayed on the home screen."
        << "\n- You can search for weather in other locations by typing the city name in the search bar." << endl;
}
void takeScreenshot() {
    cout << "\nHow to Take a Screenshot:"
        << "\n- To take a screenshot, press the 'Power' and 'Volume Down' buttons simultaneously."
        << "\n- The screenshot will be saved in the 'Gallery' or 'Screenshots' folder."
<< endl;

}

void useVoiceAssistant() {

   cout << "\nHow to Use the Voice Assistant :"

        << "\n- Activate the voice assistant by holding down the 'Home' button or
saying the voice trigger (e.g., 'Hey XYZ')."

        << "\n- You can ask the voice assistant to perform tasks like setting an
alarm, sending a message, or opening an app." << endl;

}


bool askForFurtherGuidance() {

   int choice;

   cout << "\nFor further guidance press 1, to exit press 2: ";

   cin >> choice;

   return (choice == 1);

}

int main() {

   int choice;

   do {

      displayMenu();

      choice = getUserChoice();

      if (choice == 13) break;
```

```
        switch (choice) {
            case 1: contacts(); break;
            case 2: playMusic(); break;
            case 3: openCamera(); break;
            case 4: sendMessage(); break;
            case 5: browseInternet(); break;
            case 6: setAlarm(); break;
            case 7: checkBattery(); break;
            case 8: makeCall(); break;
            case 9: sendEmail(); break;
            case 10: checkWeather(); break;
            case 11: takeScreenshot(); break;
            case 12: useVoiceAssistant(); break;
            default: cout << "Invalid choice!" << endl;
        }
    }
while (askForFurtherGuidance());

    cout << "Thank you for using the Mobile Usage Guide. Goodbye!" << endl;
    return 0;
}
```

OUTPUT:



```
                    Keypad Mobile Usage Guide
                -------------------------------
1. Contacts
2. Play Music
3. Open Camera
4. Send Message
5. Browse Internet
6. Set Alarm
7. Check Battery
8. Make a Call
9. Send Email
10. Check Weather
11. Take a Screenshot
12. Use Voice Assistant
13. Exit
-----------------------------------
Select an option for guidance: 1

How to Access Contacts :
- Open the app and navigate to the 'Contacts' tab.
- Tap on 'Add New Contact' to create a new contact.
- To search for a contact, use the search bar at the top of the screen
    .
- Tap on a contact to view details or call them directly.

For further guidance press 1, to exit press 2: 2
Thank you for using the Mobile Usage Guide. Goodbye!
```

Conclusion:

The Mobile Usage Guide program exemplifies the importance of user-centric technology by bridging the gap between mobile device functionality and user understanding. With its easy-to-use menu-driven interface, the program successfully assists users in navigating various mobile features, thereby empowering them to maximize the utility of their smartphones. Each function within the program is tailored to provide clear, step-by-step guidance that demystifies common mobile tasks, making the program a valuable tool for users of all expertise levels.

The modular structure of the code ensures ease of maintenance and scalability, allowing future enhancements to incorporate additional mobile features or adapt to new technological advancements. Furthermore, the integration of input validation and feedback mechanisms enhances the program's user experience by ensuring smooth interaction and continuous support.

In essence, the Mobile Usage Guide program fosters improved user confidence in managing smartphone capabilities, reducing frustration and enabling seamless access to essential functionalities. It stands as an example of how simple, well-designed software can make technology more approachable and user-friendly.