Project Title: Birthday Bot

Developer's Name: P. Sai Gowtham Reddy

Date of Submission: 02/06/2023

**Table of Contents**

**• Introduction: Page 3**

**• Project Overview: Page 4**

**• System Architecture: Page 6**

**• Requirements and Dependencies: Page 7**

**• Implementation Details : Page 8**

**• Error Handling and Exception Management : Page 10**

**• Future Improvements and Scalability : Page 11**

**• Summary: Page 13**

**• Resources: Page : Page 14**

**Introduction:**

The Birthday Bot project is a Python-based automated system that sends birthday wishes and images to CSV-file contacts. By automating the messaging task, the project hopes to simplify and streamline the process of sending personalised birthday greetings.

The Birthday Bot's primary goal is to eliminate the manual effort required to remember and send birthday greetings to contacts individually. The bot can automate the process and ensure timely delivery of birthday messages by leveraging the power of programming and the pywhatkit library.

It is common social practise to send birthday greetings to friends, family, colleagues, and acquaintances. However, keeping track of multiple birthdays and remembering to send birthday wishes on time can be difficult. This problem is addressed by the Birthday Bot, which automates the entire process, making it more efficient and convenient.

The bot stores contact information in a CSV file called "contacts.csv," which includes names and corresponding birthday dates. The bot can identify contacts whose birthdays match the current date by reading this file and comparing the current date to the stored birthdays.

When a match is found, the Birthday Bot sends a birthday image and a message to the contact via WhatsApp. The bot uses the functionality of the pywhatkit library to send the image and message seamlessly, ensuring that the recipient receives a personalised birthday greeting.

The bot, in addition to sending images, includes a randomization feature to provide variety in greetings. It chooses an image at random from a folder of birthday images, ensuring that each recipient receives a unique and visually appealing birthday message.

The Birthday Bot checks the current time before sending the message to account for time sensitivity. If there is enough time before the minute changes, the bot immediately sends the image and message. If the time is close to the minute change, the bot adds a delay to avoid potential problems with WhatsApp restrictions.

Overall, the Birthday Bot project provides a useful solution for automating the process of sending birthday greetings. The bot simplifies the task of remembering birthdays, sending personalised messages, and spreading joy on special occasions by leveraging Python programming and the pywhatkit library.

**Project Overview:**

The Birthday Bot project aims to automate the process of sending birthday greetings and images to contacts saved in a CSV file. This page describes the project's functionalities, technologies, and underlying CSV file structure.

**Functionality:**

The Birthday Bot's core functionality is to identify contacts whose birthdays match the current date and send them personalised birthday greetings. The bot does this by reading contact information from a CSV file, comparing stored birthday dates with the current date, and automatically sending WhatsApp messages with birthday images to the appropriate contacts.

**Technologies Used:**

The following technologies and libraries are used by the project to effectively implement its functionality:

**1. Python:** Python is the primary programming language used in the Birthday Bot's development. Python's simplicity, versatility, and extensive library ecosystem make it an excellent choice for this project.

**2. pywhatkit:** The pywhatkit library is a key component of the Birthday Bot. It provides a straightforward interface to interact with WhatsApp, allowing the bot to send messages and images to contacts automatically.

**3. pandas:** The pandas library is used to read and manipulate the CSV file that stores contact information. It provides powerful data manipulation capabilities, making it convenient to access and compare birthday dates.

**4. datetime:** The datetime module is utilized to retrieve the current date and format it according to the CSV file's date format. It allows the bot to determine whether a contact's birthday matches the current date.

**CSV File Structure:**

The project relies on a CSV file named "contacts.csv" to store the contact information. The CSV file follows a specific structure to organize the data effectively. Here is an example of the CSV file structure:

| Name | Date | Phone Number |

|------- |--------------- |----------------- |

| John | 12-03-1990 | +1234567890 |

| Emily | 05-09-1985 | +9876543210 |

| Mark | 18-06-1995 | +2345678901 |

| ... | ... | ... |

Columns in the CSV file include "Name," "Date," and "Phone Number." The "Name" column stores the contact's name, the "Date" column stores the contact's birthday in the format DD-MM-YYYY, and the "Phone Number" column stores the contact's WhatsApp phone number.

The Birthday Bot can easily access and compare birthday dates to send personalised birthday wishes by organising contact information in this structured CSV file.

The project overview explains the Birthday Bot's key features, the technologies used, and the structure of the CSV file used to store contact information. This information serves as the foundation for the subsequent pages, which delve into project implementation details and other aspects.

**System Architecture:**

The Birthday Bot project's system architecture defines the overall structure and components of the automated birthday messaging system. This article describes the architecture's key modules, interactions, and data flow.

**Primary Modules:**

**1. Data Input Module:** This module is in charge of reading the CSV file's contact information. It loads the data into memory and makes it available for further processing using the pandas library.

**2. Date Comparison Module:** This module compares previously saved birthday dates to the current date. It used the datetime module to obtain the current date and time as well as to do the necessary date comparisons.

**3. Image Selection Module:** The Image Selection module chooses a birthday image at random from a folder. This increases the variety and visual appeal of the birthday messages, guaranteeing that each contact receives a unique image.

**4. Messaging Module:** To transmit WhatsApp messages, the Messaging module communicates with the pywhatkit library. It uses the functionalities of the library to send photos and messages to the appropriate contacts. The module includes functionality to deal with timing limitations and guarantee that messages are delivered smoothly.

**Interactions and Data Flow:**

**1. Initialization:** The system begins by loading the required libraries and modules. This includes loading the CSV file into memory and importing the necessary libraries.

**2. Data Retrieval:** The Data Input module reads the CSV file and retrieves the contact information. It reads the file with the pandas library and saves the data in an appropriate data structure for further processing.

**3. Date Comparison:** This module compares the stored birthdate dates to the current date. It uses the datetime module to retrieve the current date and compares it to the dates in the contact information.

**4. Image Selection:** When a contact's birthday coincides with the current date, the Image Selection module chooses a birthday image at random from the chosen folder. This adds a surprise factor and personalisation to the birthday messages.

**5. Messaging:** To send WhatsApp messages, the Messaging module makes use of the pywhatkit package. It sends the chosen image, along with a pre-programmed birthday greeting, to the contact's WhatsApp number. To minimise potential conflicts with WhatsApp limits, the module implements timing considerations.

**6. Iteration and completion:** The system loops over the contact list, repeating the date comparison, image selection, and message processes until all relevant contacts have gotten their birthday wishes. The system confirms successful delivery and may undertake any necessary cleanup actions when the messaging process is completed.

The system architecture overview explains the essential modules and data flow of the Birthday Bot project. This understanding prepares you to delve into the detailed implementation details in the next pages.

**Prerequisites and Dependencies:**

The Birthday Bot project's success is dependent on achieving specific objectives and dependencies. This article describes the components, software, and libraries required to ensure the automatic birthday messaging system's proper operation.

**1. Python:** Python is the programming language used to create the Birthday Bot. As a result, the first need is that Python be installed on the system where the bot will be run. The project is compatible with both Python 2 and Python 3, with Python 3 being preferred due to greater capabilities and improvements.

**2. pywhatkit Library:** A critical dependency for the Birthday Bot project is the pywhatkit library. It includes the functions required to communicate with WhatsApp and send messages and images programmatically. To install the pywhatkit library, enter the following command at the command prompt or terminal:

''' pip install pywhatkit"""

3. The **pandas** library is used to read and manipulate the CSV file containing the contact information. It has useful functions for data analysis and modification. The following command can be used to install pandas:

'''pip install pandas'''

```

4. **CSV File (contacts.csv):** The project uses a CSV file named "contacts.csv" to hold contact information. The CSV file must be properly formatted, containing columns for "Name," "Date," and "Phone Number." Place the file in the same directory as the project files.

**5. Birthday photographs:** A folder containing birthday photographs is necessary to lend a personal touch **to** the birthday wishes. The folder should include a range of photos in JPG or PNG format, which will be chosen at random and sent with the messages. The photographs should be saved in a folder within the project directory called "birthday\_images."

Meeting these needs and dependencies guarantees that the Birthday Bot works properly. It is critical to ensure that the required libraries have been installed, that the CSV file has been properly structured, and that the birthday photos are available in the relevant folder. By meeting these objectives, the project will be able to function smoothly and easily send automated birthday greetings to contacts.

**Implementation Details:**

This page provides an overview of the Birthday Bot project's implementation details. It goes over the essential phases in building and implementing the automatic birthday messaging system.

**1. Establishing the Project Environment:**

- Install Python: Check to see whether Python is already installed on your system. You can get the newest version of Python from the Python website and install it by following the installation instructions.

- Install the following libraries: Install the required libraries, such as pywhatkit and pandas, using pip, the Python package installer, as described on page 6.

- Create Project Directory: Create a directory specifically for your Birthday Bot project. This directory will house your Python script, the CSV file, and the birthday image folder.

**2. Importing Required Libraries:** Import the necessary libraries, including pywhatkit and pandas, into your Python script. This allows you to use their functions and features in your code.

**3. Reading Contact Information:** Read the CSV file containing the contact information using the pandas package. Load the CSV file into a pandas DataFrame for easy data access and manipulation.

**4. Date Comparison and Message Sending Logic: -** Using the datetime module, iterate over the rows of the DataFrame and compare the recorded birthday dates with the current date. Determine the contacts whose birthdays coincide with the current date.

- Send WhatsApp messages with birthday photos to the specified contacts using the pywhatkit module. Using the Image Selection module, select an image at random from the selected folder and send it along with a pre-defined birthday message.

**5. Handling Timing Constraints: -** When sending messages, keep the current time in mind to avoid conflicts with WhatsApp limits. Check the current time with the datetime module and, if necessary, add a delay to assure message delivery.

**6. Iteration and completion:** - For each contact whose birthday matches the current date, repeat the date comparison and message processes. Iterate through the DataFrame until all appropriate contacts have received birthday messages.

- After completing the messaging procedure, provide feedback or display a success message to indicate that the messages were successfully sent.

**7. Testing and refining:** Run the Python script to test the Birthday Bot and ensure that the messages and images are being sent successfully.

- Debugging and fine-tuning the code as needed to solve any bugs or errors discovered during testing.

**8. Starting the Birthday Bot:** - Run the Python script to start the Birthday Bot. Check that the CSV file and the folder containing the birthday photographs are in the proper places.

You can successfully develop and execute the Birthday Bot project if you follow these implementation specifics. Testing and improving the code is critical to ensuring that the bot works properly and delivers automated birthday greetings to the contacts in your CSV file.

**Error Handling and Exception Management :**

It is critical to integrate proper error handling and exception management in every software project to ensure the system's robustness and reliability. This page focuses on the strategies and practises used in the Birthday Bot project to handle faults and exceptions.

**1. Input Data Validation:** Ensure the integrity and validity of the contact information by performing data validation checks on the CSV file. Check that the necessary columns are present and in the anticipated format.

- Examine the CSV file for any missing or improper values, such as empty names or wrong date formats. Handle these scenarios correctly by ignoring the incorrect entries or displaying error notices.

**2. Error Handling Mechanisms:** - Use try-except blocks to catch and manage exceptions that may occur during programme execution. Wrap crucial areas of code with appropriate try-except blocks, such as file reading and message sending activities.

- Recognise and handle any exceptions that may occur, such as file reading faults or network connection problems. To assist in troubleshooting, display informative error messages to the user.

**3. Logging and Error Reporting:** - Implement logging tools to capture pertinent error and exception information. To log error information, timestamps, and any other relevant data, use Python's logging module.

- Error reports should be generated that summarise the errors encountered, including the nature of the error, the affected contacts, and any error codes or messages related with the failures. These reports can help with troubleshooting and future enhancements.

**4. Graceful Termination:** Ensure that in the event of critical errors or exceptions, the programme terminates gracefully. - If necessary, release any acquired resources, close files, and clean up temporary data. Display relevant messages or prompts informing the user of the termination and any necessary steps.

**5. useful and User-Friendly Error Messages:** Create useful and user-friendly error messages to guide users when errors occur. Communicate the nature of the error, potential causes, and suggested remedies or steps to remedy the issue in a clear and concise manner.

- Consider including instructions for updating or correcting the CSV file, checking network connectivity, and dealing with any other frequent error scenarios that may occur during execution.

The Birthday Bot project can handle unexpected events gently and provide useful feedback to users when failures occur by implementing robust error handling and exception management practises. This method improves the system's stability and usability, resulting in a better user experience and easier troubleshooting.

**Future Improvements and Scalability:**

The Birthday Bot project establishes a solid framework for automating birthday greetings and message delivery. However, there are numerous possibilities for future improvements and scalability. This page investigates prospective areas for project enhancement and expansion.

**1. User Interface:** Create a UI for the Birthday Bot that allows users to quickly maintain contacts, check birthday alerts, and customise messages or picture choices. A user interface (UI) can improve the user experience and make the project more accessible to non-technical people.

**2. Scheduling:** Implement scheduling capabilities to automate the Birthday Bot's execution at certain times or intervals. This enables users to programme the bot to deliver birthday messages automatically, providing a totally hands-free experience.

**3. personalisation:** Improve personalisation by allowing users to customise birthday messages or create templates for various contacts. Allow the contact's name to be included in the message, as well as personal anecdotes and emojis for added flair.

**4. Support for Other Messaging Platforms:** Extend the project's support for messaging platforms other than WhatsApp. Include APIs for systems like Facebook Messenger or Telegram, allowing users to send birthday greetings over several channels.

**5. picture Selection Enhancement:** Enhance the picture selection module by including image recognition or sentiment analysis methods. This would allow the bot to choose photos depending on the contact's interests, hobbies, or mood, personalising the birthday wishes even more.

**6. Error Handling and Logging:** Improve error handling and logging mechanisms to provide more detailed error reports and automatic error resolution recommendations. This makes it easier for users to troubleshoot problems and lowers the need for manual intervention.

**7. Internationalisation:** Add internationalisation features to support many languages and date formats. Allow users to choose language preferences and customise the bot's messaging accordingly.

**8. Scalability:** Optimise the code and system architecture to efficiently handle larger contact lists. To boost performance and scalability, use techniques such as parallel processing or distributed computing.

**9. Contact Management System Integration:** - Connect the Birthday Bot to popular contact management systems like Google Contacts or Microsoft Outlook to automatically import contact information and synchronise birthday data.

**10. Automated Reminders:** - Extend the bot's capabilities to send automated birthday reminders to users, ensuring that important birthdays are never missed.

The Birthday Bot can expand into a more feature-rich and adaptable tool for automating birthday wishes and communications if these future additions and scaling possibilities are considered. Continuously refining and growing the project assures its long-term relevance and utility.

**Summary:**

The Birthday Bot project is a creative and useful method for automating birthday greetings and message delivery. The project automates the process of sending personalised birthday messages to contacts stored in a CSV file by harnessing the capabilities of Python and tools like as pywhatkit and pandas. We have discussed several parts of the project in this overview, including the system architecture, requirements, implementation details, error handling, and potential upgrades.

The Birthday Bot has various perks and benefits. It eliminates the need for manual intervention in sending birthday greetings, saving users time and effort. The integration with WhatsApp enables quick and easy communication with contacts. Birthday graphics give a personal touch and visual appeal to the greetings, making them more memorable for the recipients. The project is intended to handle big contact lists and may be easily customised and modified to meet the needs of the user.

The Birthday Bot, like every software project, has potential for improvement and additional development. Features such as a user interface, scheduling capabilities, greater personalization choices, support for more messaging platforms, and interaction with contact management systems could be added in the future. The Birthday Bot may become even more versatile, user-friendly, and scalable by adopting these additions.

Finally, the Birthday Bot project exemplifies the power of automation and technology in automating and simplifying repetitive operations. It gives delight and convenience to both the giver and the recipient by sending automatic birthday wishes and messages. We can unleash even more potential in automating numerous elements of our lives if we continue to enhance and expand on this project.

**Resources**

Various resources were used to obtain knowledge, learn new techniques, and implement the necessary functionalities during the Birthday Bot project's development. This page serves as a resource list for the sources used in the development of this project.

1. Python Documentation: The official Python documentation provides useful information on syntax, libraries, and recommended practises. You may find it at <https://docs.python.org/>.

2. pywhatkit Documentation: The pywhatkit library documentation aided in the incorporation of WhatsApp messaging capabilities into the project. It went into great length about the library's functions and usage. You may find it at <https://github.com/Ankit404butfound/PyWhatKit>.

3. pandas documentation: To understand how to read and manipulate data from CSV files, the pandas library documentation was consulted. It demonstrated and explained several pandas functions. Pandas is available at https://pandas.pydata.org/docs/.

4. Stack Overflow: The prominent programming Q&A website, Stack Overflow, was a great resource for diagnosing specific issues and discovering answers to common development challenges. You can find it at https://stackoverflow.com/.

5. PythonPackageIndex (PyPI): PyPI was used to discover and install the project's required libraries, such as pywhatkit and pandas. It is the official Python package repository. AvailableAvailable at: https://pypi.org/

6. Microsoft Word material: To understand how to make a table of contents and format headings, the official Microsoft Word material was consulted. It gave examples and step-by-step directions. You can find it at: https://support.microsoft.com/en-us/word

7. Online Python Tutorials: A variety of online tutorials and manuals were used to understand various Python programming techniques and topics. Real Python (https://realpython.com/) and GeeksforGeeks (https://www.geeksforgeeks.org/) were especially useful.

8. own Experience: The project developers' own experience and knowledge influenced the design and implementation of the Birthday Bot. Their knowledge of Python programming, data manipulation, and WhatsApp integration all contributed to the project's success.

Please keep in mind that the references provided above are merely for informational reasons. When working on similar projects, it is critical to study official documentation and acceptable sources for accurate and up-to-date information.