

Automated Parade State Program

**Prepared by
Keagan Suah**

11 May 2022

Other Information

The project has been posted online into Github, where developers contribute their ideas to the open-source community. The Python code file can be found inside. You may click on this link to view my project:

<https://github.com/KeaganSuah/Automated-Parade-State-Program>

Contents

1 Introduction

1.1 Problem Statement

1.2 Project Objectives

2 Modules & Data Structures

2.1 Modules

2.2 Data Structures

3 Database

3.1 Adding Personnel in Database

3.2 Removing personnel from Database

3.3 Database Functions

4 User Interaction

5 Displaying Parade State

6 Main Operator

7 Conclusion

7.1 Future Enhancement

1. Introduction

1.1 Problem Statement

In the military, personnel must submit their attendance every morning before working hours to determine whether or not they are present; this is referred to as a parade state. The Parade state includes the personnel rank and name, as well as their current status. It also includes a parade state summary, which shows the number of people with the same status. The procedure for submitting a parade state begins with each individual being responsible for submitting their parade state in a Whatsapp chat group. Others will gradually copy the previous message and send the message with the status from the same row as their information. Following the submission of all personnel, a clerk will collect all of the data and transfer it to an excel sheet. The excel sheet will include the parade state from Whatsapp as well as a calculated summary of it. They will then submit their work to the manpower branch for review. Other interested parties, such as the company sergeant major and the commanding officer, may also view the sheet to determine the number of people present at the time.

1.2 Project objectives

Despite the fact that this practice has been going on for quite some time, it raises a number of issues that cause inconvenience for others. Personnel must copy and paste previous messages as well as clean up the data before submitting the new status. Many people may want to submit their parade state all at once, which may overwrite some messages sent by others, resulting in an inaccurate parade state. Another issue is that the whatsapp group, which is supposed to be for work-related discussion, is now a place for personnel to submit their status, resulting in a flood of messages. It takes the clerk a long time to manually type the current parade state into an excel sheet and calculate the parade state summary. In some cases, clerks may be too busy and forget to submit the parade state, forcing the manpower branch to go after them.

The commanding officer intends to digitalize many of these everyday processes, the most important of which is the parade state. That is why I developed this Automated Parade State program, which can address this issue. Every morning, employees will post their status on Google forums, which allows multiple users to post at the same time. The Forums data is linked to the Google sheet. This program will then extract these data from Google Sheets, authenticate the personnel data, and clean the raw dirty data, which may contain duplicates or unrecognizable personnel. After that, the clean data will be formatted and structured into a table format. The program's functions will assist in using the table and generating the parade state as well as the parade state summary, which is updated daily.

The Automated Parade State program addresses the commanding officer's digitalization plan for his unit. The program also makes it easier for personnel to submit their parade states while also reducing the workload that clerks are responsible for doing every morning. Commanders who are interested in the daily parade state, such as the company sergeant major, can easily view it because it is designed in a properly formatted table.

2. Modules & Data Structure

Importing Modules

Integrated Development Environment

```
1 import gspread
2 from oauth2client.service_account import ServiceAccountCredentials
3 from tinydb import TinyDB, Query
4 from prettytable import PrettyTable
5 from datetime import date
```

These are the critical modules that must be imported for the program to function properly. "gspread" and "oauth2client.service account" are used by the program to pull data from Google Sheets into our program. "tinydb" is a database module that allows us to keep small amounts of data in a Json file. The "prettytable" module is used to help generate our output in the form of a properly formatted table, and the "datetime" module is used to generate our current date.

Data Structures

Integrated Development Environment

```
29 """DATA STRUCTURES"""
30 """ Dictionary will be used to store data for the name and their statuses. Choosing Dictionary also allow this
31 system to remove any duplicated names. Users that keyed the same name twice but with a different status, will
32 update his status with the latest submitted one """
33 current_dict = {}
34
35 # The pretty table module that display the parade state in a properly formatted table
36 ps_Table = PrettyTable(["No.", "Personnel", "status"])
37 sum_Table = PrettyTable(["Status", "Total Number"])
38 """ This helps to format the raw data into a understandable dictionary called current_dict.
39 It is done by getting data from the first and second question """
40 for personnel in data:
41     current_dict[personnel.get('Full Name (Name keyed in the database)')] = personnel.get('Status')
42
43 # This list will be used to display the parade state
44 display = []
45 # dictionary helps to provide the summary of the parade state
46 present_day = {"Present": [], "Work From Home": [], "Outside Stationed": [], "Attached Out": [],
47               "On Course": [], "Day Off": [], "Local Leave": [],
48               "Overseas Leave": [], "Medical Leave": [], "Medical/Dental Appointment": [],
49               "Report Sick Inside": [], "Report Sick Outside": [],
50               "Hospitalized / Sickbay": [], "AWOL": [], "OTHERS": []}
```

These are the data structures that will assist us in storing and presenting data for our program. The "current dict" aids in the storage of personnel rank, name, and status data. By selecting dictionary, the system will remove any duplicated names. If a user enters their status again, the sheet will be updated with the most recent status. A function is also written to format the raw data from the Google Sheet into the dictionary "current dict."

The "ps table" and "sum table" data structures are from the earlier imported module. The first row in these data structures represents the name of the variables in each column. Following the completion of some functions, the personnel information and status will be stored in these two data structures.

The other data structures, "display" and "present day" are used to help us present the parade state and its summary on the present date. These data structures aid in categorizing data and removing irrelevant information. The data that the variables receive will be saved first and will be used later. These data structures will be used by other functions to help generate the parade state.

3. Database

A database is created to store the personnel's information and details, as well as to allow the program to authenticate the names entered in the Google forums. This is significant because people may make human errors and enter their name incorrectly into the system, or other people who no longer belong in this unit may enter their name as well. This database will help to prevent such incidents by allowing only personnel from this unit to access the program.

Adding Personnel into Database

Output

```
Add or Remove personnel in Database/Submit Parade State (D/S): 0
(Add/Remove) from database?: Add
what's your name: oliver
what's your rank: me3
Do you still want to add?(Yes/No): Yes
what's your name: william
what's your rank: lcp
Do you still want to add?(Yes/No): Yes
what's your name: jones
what's your rank: lcp
Do you still want to add?(Yes/No): No
Add or Remove personnel in Database/Submit Parade State (D/S):
```

Json File

```
Parade State.py x dbjson x parade-state-350612-4de42593bc63.json x
{"_default": {"1": {"rank": "cpl", "name": "keagan"}, "2": {"rank": "me5", "name": "kevin"},
"3": {"rank": "me3", "name": "liam"}, "4": {"rank": "me3", "name": "noah"}, "5": {"rank": "me2", "name": "oliver"},
"6": {"rank": "lcp", "name": "william"}}}
```

The output shows how it will interact with the user, asking if they want to add or remove personnel from the database. If they select add, the program will prompt them for the personnel's name and rank. Another question will be asked of the user, asking if they want to add another person, allowing the user to enter multiple personnel.

As shown above, the Json file acts as a database to store the personnel status. The names that the user has keyed into the output will be placed inside the database within the Json file.

Removing Personnel from the database

Output

```
Add or Remove personnel in Database/Submit Parade State (D/S): 0
(Add/Remove) from database?: Remove
what's your name: keagan
Add or Remove personnel in Database/Submit Parade State (D/S): 0
(Add/Remove) from database?: Remove
what's your name: kevin
Add or Remove personnel in Database/Submit Parade State (D/S): |
```

Json File



```
{
  "_default": {
    "3": {
      "rank": "me3",
      "name": "liam"
    },
    "4": {
      "rank": "me3",
      "name": "noah"
    },
    "5": {
      "rank": "me2",
      "name": "oliver"
    },
    "6": {
      "rank": "lcp",
      "name": "william"
    }
  }
}
```

If they select remove, the program will prompt them to enter the personnel name they wish to remove from the database. The Json file, which serves as a database, will respond appropriately by removing the personnel that the user has entered. In this case, the user chose to delete "keagan" and "kevin." The Json file acted in accordance with the user's input and removed all personnel details. The database deletes all personnel details such as rank and name.

Database Functions

Integrated Development Environment

```
53 """DATABASE FUNCTIONS"""
54
55
56 # function to enter details into the database
57 def insert(rank, name):
58     db.insert({'rank': rank, 'name': name})
59
60
61 # function to remove a personnel details from the database
62 def delete_by_name(name):
63     db.remove(User.name == name)
64
65
66 # Function ask user to enter the details of the personnel to add into the database
67 def add_database():
68     name = str(input("what's your name: "))
69     rank = str(input("what's your rank: "))
70     insert(rank.lower(), name.lower())
71
72
73 # Function ask user to enter the name of the personnel to remove from database
74 def remove_by_name():
75     name = str(input('what's your name: '))
76     delete_by_name(name.lower())
```

These are the functions that assist the program in adding and removing personnel from the database. To insert or delete personnel from the database, use the "insert" and "delete by name" functions. These functions will use variables called name and rank to add or remove personnel that the user has keyed.

The remaining functions, "add database" and "remove by name," interact with users by asking them to enter the personnel's name or rank. The functions will convert all of the information entered by the user and change all of the letters to lower case. This is required because a mix of lowercase and uppercase letters will confuse the program. Because there is no consistency, the program may misidentify a person with an uppercase and lowercase name as two different people. The keyed detail will then be assigned to the variables that will be used for the functions listed above.

4. User Interaction

Integrated Development Environment

The image shows a mobile app interface for submitting parade status and a corresponding Google Sheet. The app interface on the left includes a login section with the email 'viderpergames@gmail.com', a 'Draft saved' indicator, and a 'Full Name (Name keyed in the database)' field containing 'lucas'. Below this is a 'Status' section with radio buttons for 'Present' (selected), 'Work From Home', 'Outside Stationed', 'Attached Out', 'On Course', 'Day Off', 'Local Leave', 'Overseas Leave', 'Medical Leave', 'Medical/Dental Appointment', 'Report Sick Inside', 'Report Sick Outside', 'Hospitalized / Sickbay', 'AWOL', and 'OTHERS'. A 'Submit' button is at the bottom. To the right, a Google Sheet titled 'Parade state' displays the submitted data in a table with columns for 'Timestamp', 'Full Name (Name keyed)', and 'Status'.

	A	B	C
1	Timestamp	Full Name (Name keyed)	Status
2	28/05/2022 01:12:17	lucas	Present
3	28/05/2022 01:12:51	keagan	Work From Home
4	28/05/2022 01:13:10	liam	Work From Home
5	28/05/2022 01:13:24	noah	Local Leave
6	28/05/2022 01:13:34	oliver	On Course
7	28/05/2022 01:13:45	william	Hospitalized / Sickbay
8	28/05/2022 01:13:57	james	On Course
9	28/05/2022 01:14:11	benjamin	Present
10	28/05/2022 01:14:20	lucas	Report Sick Outside
11	28/05/2022 01:14:35	henry	Outside Stationed
12	28/05/2022 01:14:47	ethan	Present
13	28/05/2022 01:14:55	daniel	Present
14			
15			
16			
17			
18			

From the above image, this is how personnel will submit their parade status. This program allows personnel to help key in their parade state while using the convenience of their phone. This is more convenient for them because they can submit it whenever they want. They also do not have to flood the Whatapp Group with parade state submissions, which is the current procedure.

The data from the Google Forum will be sent to Google Sheets and updated as needed. The Google forum and the Google sheets are both linked. As seen above in the Google Sheets, there is a duplicate name within the sheet, as well as a personnel that we had previously removed from the database. This demonstrates the importance of having data structures and a database in order to remove these duplicates. The data structure aids in the removal of duplicates and uses the most recently submitted data as the parade state status. The database aids in the authentication of personnel by matching the data in Google sheets with the records in the database. This aids in the removal of any unwanted personnel recorded in the Google sheet.

Integrated Development Environment

```
7  """IMPORT FROM GOOGLE SHEETS"""
8
9  # List provides the necessary links to connect to the API
10 scope = ["https://spreadsheets.google.com/feeds", 'https://www.googleapis.com/auth/spreadsheets',
11          "https://www.googleapis.com/auth/drive.file", "https://www.googleapis.com/auth/drive"]
12
13 # Function opens the Json file named parade state
14 creds = ServiceAccountCredentials.from_json_keyfile_name("parade-state-350612-4de42593bc63.json", scope)
15
16 # This give authority for using the file and opening the API
17 client = gspread.authorize(creds)
18
19 # Opens the google-docs file named 'Parade State'
20 sheet = client.open("Parade state").sheet1
21
22 # Gets all the data in the Google Sheets
23 data = sheet.get_all_records()
24
25 # The variables needed when managing and importing the Tiny Database
26 User = Query()
27 db = TinyDB('db.json')
```

These functions aid in the assignment of variables and the extraction of data from Google Sheets into the program. The "scope" consists of the links that will be used by the program to edit the Google sheets. The "creds" variable is used to authenticate the user by using the Google account that was assigned when the API for the Google sheet was created. The Google account address is stored in a Json file, along with other credentials that aid in the data extraction process. The "client" variable then authorizes the previous variable using a module. The variables "sheet" and "data" are then used to open the extracted data and retrieve all of the records entered by all personnel. Other functions will then collect all of this data and store it in the data structures that will be used later.

5. Displaying Parade State

Output

```
Add or Remove personnel in Database/Submit Parade State (D/S): $ View Parade State or Summary (P/S): $
Parade State has been submitted
Parade State Summary2022-05-28
View Parade State or Summary (P/S): P
Parade State 2022-05-28
Total Present: 3/10
+-----+-----+
| No. | Personnel | status |
+-----+-----+
| 1 | CPL LUCAS | Report Sick Outside |
| 2 | ME3 LIAM | Work From Home |
| 3 | ME3 NOAH | Local Leave |
| 4 | LCP WILLIAM | Hospitalized / Sickbay |
| 5 | LCP JAMES | On Course |
| 6 | CPL BENJAMIN | Present |
| 7 | ME5 HENRY | Outside Stationed |
| 8 | ME4 ETHAN | Present |
| 9 | PTE DANIEL | Present |
| 10 | ME2 OLIVER | On Course |
+-----+-----+
View Parade State or Summary (P/S):
```

```
View Parade State or Summary (P/S): $
Parade State Summary2022-05-28
+-----+-----+
| Status | Total Number |
+-----+-----+
| Present | 3 |
| Work From Home | 1 |
| Outside Stationed | 1 |
| Attached Out | 0 |
| On Course | 2 |
| Day Off | 0 |
| Local Leave | 1 |
| Overseas Leave | 0 |
| Medical Leave | 0 |
| Medical/Dental Appointment | 0 |
| Report Sick Inside | 0 |
| Report Sick Outside | 1 |
| Hospitalized / Sickbay | 1 |
| AWOL | 0 |
| OTHERS | 0 |
| TOTAL STRENGTH: | 7 |
| TOTAL ABSENTEES: | 3 |
+-----+-----+
```

This is the main output in which users are most interested. It assists in displaying and formatting the parade state, as well as the parade state summary, for the user to view. The modules we imported, such as "prettytable" and "datetime," assist in formatting and structuring the parade state in a more appealing manner for the user. When data is formatted in this manner, the user can properly view and interpret it. The table also includes useful information for the user, such as the rank, name, status, and total number of people with a similar status.

Integrated Development Environment

```
79  """SUBMITTING AND CALCULATING PARADE STATE FUNCTIONS"""
80
81  def submit():
82      number = 1
83      for key, value in current_dict.items():
84          result = db.search(User.name == key)
85          for match in result:
86              ps_Table.add_row([str(number), str(match.get('rank')).upper() + ' ' + match.get('name').upper(),
87                               str(value)])
88              present_day[value] += [match.get('name')]
89              number += 1
90
91
92  def total_strength():
93      strength = 0
94      absent = 0
95      item = list(present_day.keys())
96      for word in item[0:5]:
97          strength += len(present_day.get(word))
98      for word in item[5:]:
99          absent += len(present_day.get(word))
100      sum_Table.add_row(['TOTAL STRENGTH: ', str(strength)])
101      sum_Table.add_row(['TOTAL ABSENTEES: ', str(absent)])
102      print(sum_Table)
```

These functions assist in transferring data from data structures into "prettytable" variables, which produce the parade state in a table format. As previously stated, the data structures that hold all of the extracted data will be used in this function. The "submit" function extracts the data from the data structure and stores it in the "ps Table" variable. It also displays the title and the current date.

The parade state summary is generated by the "total strength" function. It first extracts all possible user statuses and stores them in the "sum table" variable. It also counts the number of people who submitted the same status. All of this contributes to the parade state summary.

Integrated Development Environment

```
105 """DISPLAYING FUNCTIONS"""
106
107
108 def display_ps():
109     print('Parade State ' + str(date.today()) + "\nTotal Present: " + str(len(present_day['Present']))+'/' +str(len(db)))
110     print(ps_Table)
111
112
113 def display_sum():
114     print('Parade State Summary' + str(date.today()))
115     for key, value in present_day.items():
116         sum_Table.add_row([str(key), str(len(value))])
117     total_strength()
```

Both the "display ps" and "display sum" functions will print both the "ps table" and "sum table" data structures. Another feature of these functions is that they output the parade state's title and current date. Because this must be done every working day, the process of changing the date must be automated.

6. Main Operator

Integrated Development Environment

```
122 while True:
123     question = str(input('Add or Remove personnel in Database/Submit Parade State (D/S): '))
124     if question == 'D':
125         edit = str(input('(Add/Remove) from database?: '))
126         if edit == 'Add':
127             while True:
128                 add_database()
129                 ask = str(input('Do you still want to add?(Yes/No): '))
130                 if ask == 'No':
131                     break
132             elif edit == 'Remove':
133                 remove_by_name()
134         elif question == 'S':
135             submit()
136             print('Parade State has been submitted')
137             while True:
138                 second = str(input('View Parade State or Summary (P/S): '))
139                 if second == 'P':
140                     display_ps()
141                 elif second == 'S':
142                     display_sum()
143                 else:
144                     break
145     else:
146         last_check = str(input('Before leaving, would you like to deleted today\'s parade state? (Y/N): '))
147         if last_check == 'Y':
148             count = len(data)
149             sheet.delete_rows(2, count + 1)
150             print('Parade State deleted')
151             print('Thank you and have a nice day')
152             break
153         else:
154             if last_check == 'N':
155                 print('Thank you and have a nice day')
156                 break
157
```

The main operator, like the program's backbone, connects all of these functions and processes the program for the user. This consists of many inputs that will prompt the user with questions, such as asking if they want to edit the database or submit the parade state. Depending on the user input, the program will act in accordance with what the user requested, utilizing the relevant functions to aid in the process and generating what the user desires. At the end of the program, before the user exits, it will ask the user if he or she wants to delete the data in Google Sheets. This is critical because the Google sheets must be cleared before the next day, as personnel will be submitting their new parade state the next day.

7. Conclusion

The automated Parade State allows personnel to submit their parade state from the convenience of their own phone. The program prevents them from flooding the Whatsapp message, which is supposed to be used for work-related discussions rather than sending parade state. It can also display the parade state in an appealing table format to the user. The summary parade state will also be calculated automatically, saving the user time and allowing the user to quickly check the parade state and know the attendance in his unit. However, there are some areas where the program can be improved to provide a higher quality of service to the user and provide a more enjoyable experience.

Future Enhancements

Following the presentation of the project to the interested parties, some feedback was given on how they would like the program to develop. Those changes that could be made quickly and easily have been made. However, some require more time to adjust and improve before being presented to the client again.

According to the interested parties, the program still requires a computer to run it, which was not applicable. This raises two major concerns about the current program. The first issue is that, because this program is intended for military use, there are numerous limitations when it comes to using the military computer to generate and display the parade state. The second issue is that, because a computer is still required, the company sergeant major may have difficulty using it. He also mentioned that it would be preferable if he could check the parade status from his phone or while on the go.

This sparked the idea of developing a website that can be accessed via a smartphone. This is more convenient because the company sergeant major can now view the parade state without using a computer. A website module such as "Flask" can be used to automate the parade state program that helps to create the website as well as program the website's back-end. Because the interested parties' primary focus is on displaying the parade state conveniently, the front-end is not required at this time.