The Army Battalion Website

Created by

Keagan Suah

12 January 2023

Other Information

The project has been made into a portfolio where you can have a demonstration of how the project works. You may click on this link to view my project:

http://keagans.duckdns.org/portfolio

Contents

1 Introduction

1.1 Problem Statement

1.2 Learning the Skills

2 Website's Features and Functions

2.1 Security & Access

2.2 Daily Refresh

3 Database

3.1 Attendance Taking System

3.2 Users Account

3.3 Duty Roster System

4 Data Transfer to Excel

4.1 How it works

4.2 Attendance Taking System

4.3 Duty Roster System

5 Automated Attendance Taking

5.1 Individual Submission

5.2 Commander's Attendance Taking

6 Automated Duty Roster

6.1 Individual Submission

6.2 Duty Roster Algorithm

6.3 Managing Duty

7 Conclusion

7.1 Future Enhancement

7.2 Testimonial Received

1. Introduction

1.1 Problem Statement

I discovered many daily procedures that cause inconvenience and waste of manpower after serving in the Singapore Armed Forces for over a year. Seeing this as an opportunity, I decided to create a system that can digitalise and automate all of these procedures. My two main automations are the Automated Attendance Taking system and the Automated Duty Roster system.

Before the automation systems were created, the procedure of performing the two tasks was inefficient. The first task is the submission of attendance, also known as parade state. The parade state submission process begins with individuals submitting their parade state into a Whatsapp chat group. Following the submission of all personnel, a clerk will collect all of the data and transfer it to an excel sheet. Traditional attendance-taking has the disadvantage of being messy, inefficient, and inaccurate. Many of these servicemen also do not have access to a desktop, making them travel out of their workshop and into the headquarters to submit their parade state.

The second task is to plan and manage army duties. Due to the numerous steps involved, it usually takes a week for an officer or sergeant major to produce the duty roster. As a result of traditional duty planning, many of these servicemen were dissatisfied with the roster. As it is done manually, there will be human errors and biases.

1.2 Learning the Skills

I needed to learn more programming languages and frameworks, as well as how to build a website server before I could create this project. I learned Javascript, HTML, and CSS for the frontend of the website, while Python and many of its frameworks assist in the making of the backend of the website, and algorithms.

The most important framework I needed to learn was Flask. Flask is a Python web framework that includes many other necessary extension frameworks, such as flask SQLAlchemy and flask login. As the website's foundation, I must learn this framework and all of its extensions. It can generate and redirect URL pages, as well as render its HTML template. Data structures and variables from the backend can also be rendered and transferred to the frontend. The other frameworks I've learned aid in the development of algorithms and system goals.

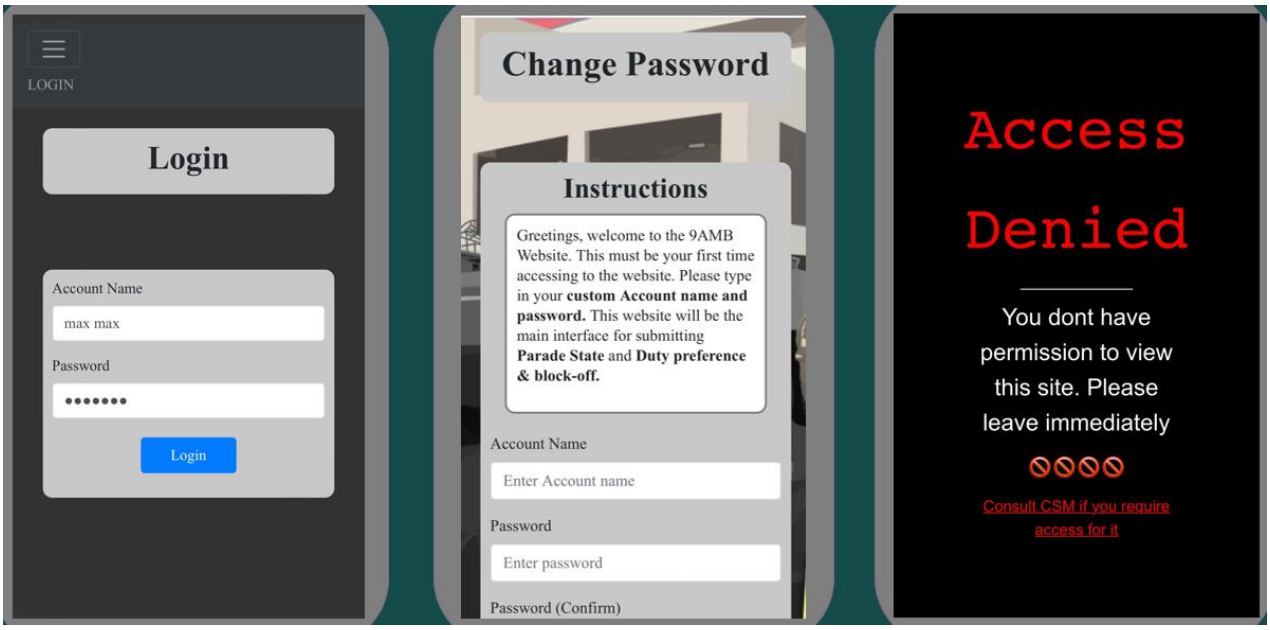
Cloud computing is the final topic I needed to learn. The systems must be accessible to the entire battalion, so cloud computing allows them to connect their devices to the website's virtual server. Amazon web services are the cloud provider I've chosen because it offers more services than other cloud providers. When comparing AWS to Google Cloud Platform, for example, AWS provides significantly more services than GCP. The services available on AWS are extremely diverse. These various services are extremely well integrated, resulting in a very comprehensive cloud service.

Virtual servers and domain names are two examples of these services that I've learned about. I learned how to use AWS and chose AWS Lightsail as the website's server. Lightsail provides easy-to-install virtual servers backed by AWS's power and reliability. To improve the website's performance, I decided to run the server on Linux, the most popular operating system for web servers, which results in more features for web designers. Hence, this help to ease my learning and works well with the project I’m creating.

2. Website Features & Functions

Security & Access

User Interaction



Flask Login, a framework for security and access, safeguards the website and protects data from being accessed by the public. The framework restricts the general public from accessing the website without having a user account. This module has a login required function that can be placed on pages, redirecting traffic to the login page if they are not logged in. An account is created whenever a new serviceman enters the battalion. The website will then use SQLalchemy, a database engine to transfer the serviceman's particulars and role-based-access control into the website.

Role-based access control is a mechanism that restricts system access is a feature within the website. It entails assigning permissions and privileges to authorised users. Using role-based access control allows servicemen to have varying levels of access based on their roles and responsibilities. This safeguards sensitive data and ensures that servicemen can only access information and perform actions required to carry out their duties. If the user does not have the role-based access control to access certain pages, they are directed to the access denied page.

Daily Refresh

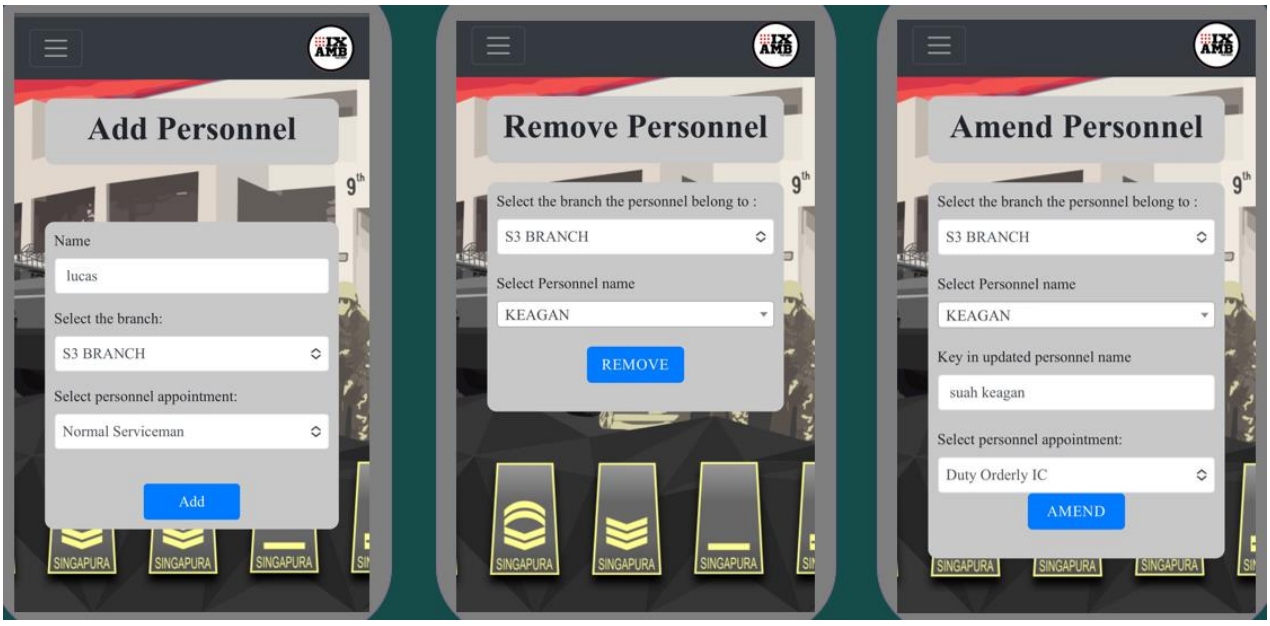
The website contains a digital attendance system and a digital duty roster system, both of which require certain procedures to be refreshed daily or monthly. The Python Framework APScheduler and Datetime are used here to assist the two systems in automating functions to run on specific dates and times throughout the month. At the start of each month, the Excel sheet that stores the previous month's data for the digital attendance system and digital duty roster system will be refreshed.

3. Database

A database is required to store a large amount of data, including all the serviceman's data and their role-based access control for the Attendance System and Duty Roster System. Flask-SQLAlchemy is a database engine that helps using SQLAlchemy with Flask easier. The database used by the website is SQLite. SQLite is a database for the majority of low to medium-traffic websites. SQLite is the best database for this website because the project's traffic is low. Both systems share the same database to ensure the data shared among both systems is consistent and accurate

Attendance Taking System

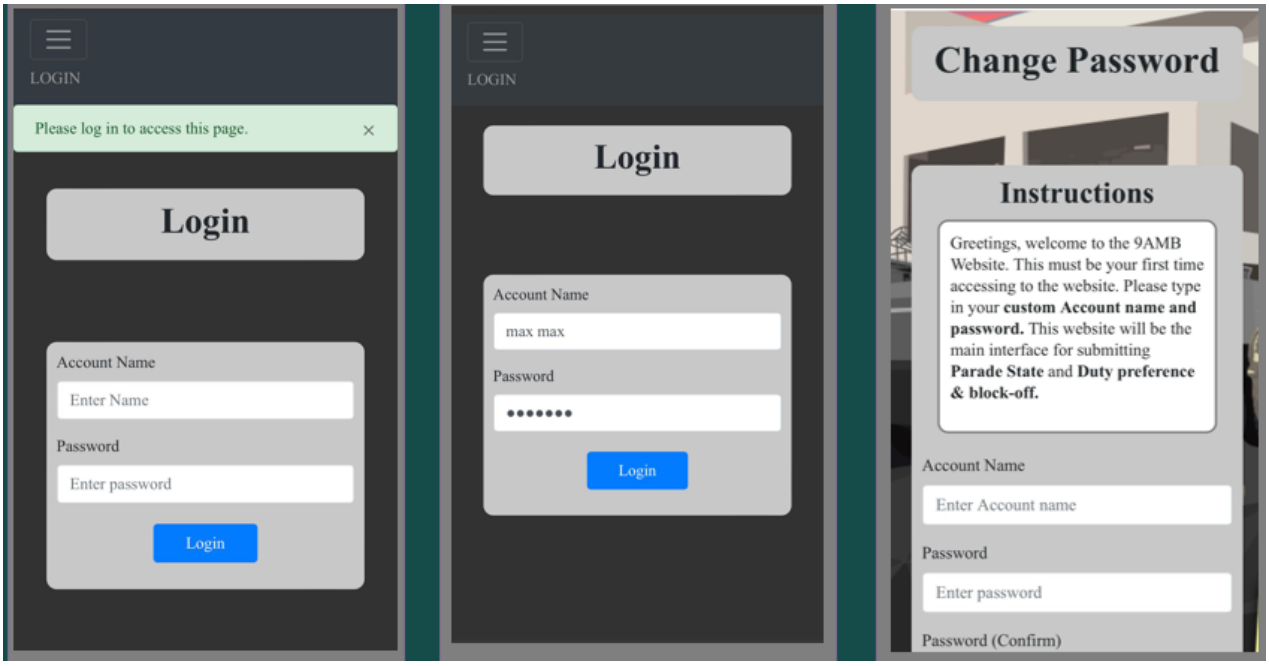
User Interaction



Every serviceman in the battalion is profiled in the database. The battalion is divided into numerous workshops and branches due to its size. When a serviceman joins, leaves, or advances in rank, the website notifies the database. The database will receive the data from the backend system whenever a new entity is added or removed, or an existing entity is amended. The three pages shown above are the website's user interface for entering data.

User Account

User Interaction

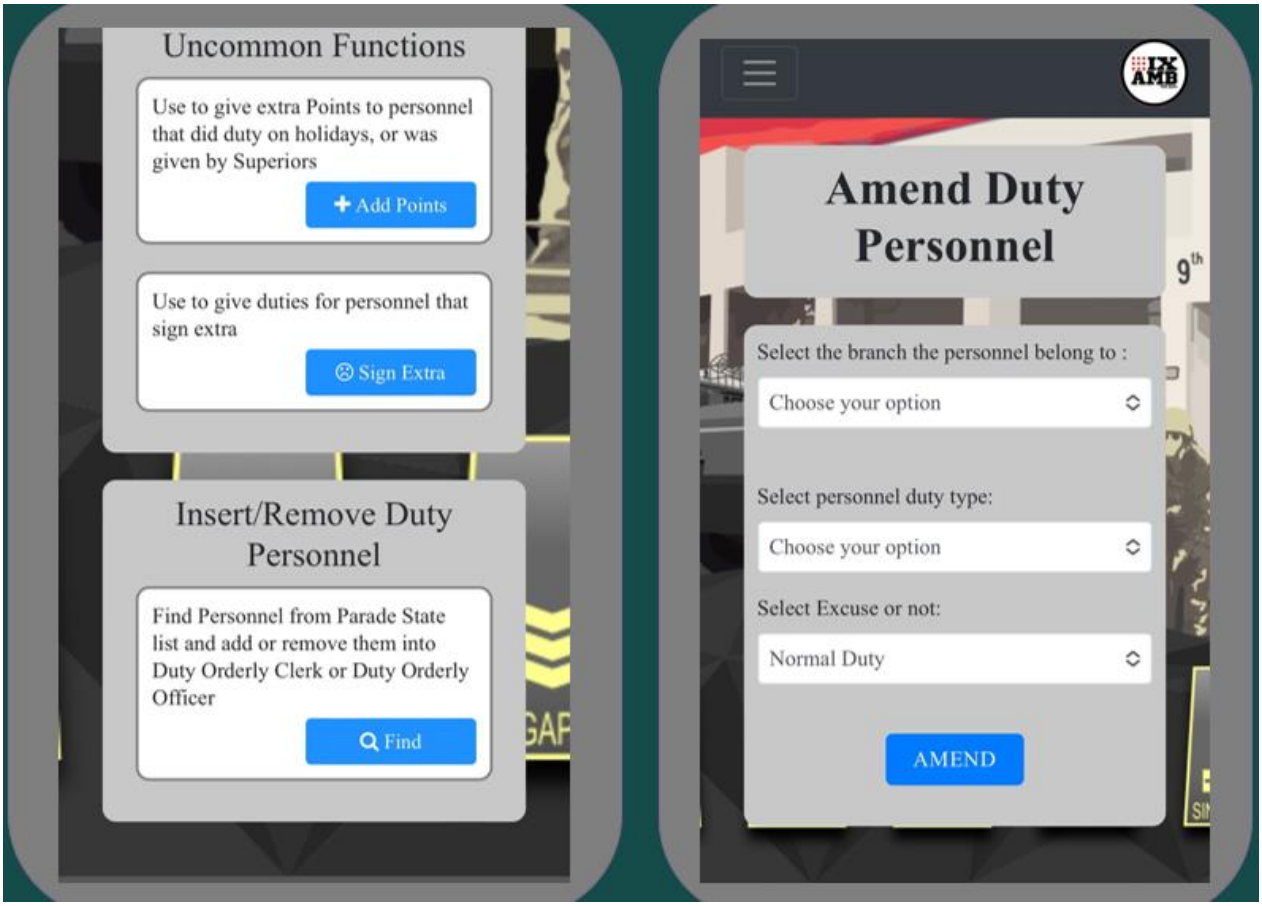


When the new serviceman is added to the database, a new account is created. The serviceman credentials, which include their shortened form name, password, role-based access control, and duty type will be added to the database. New users will be required to change their account name and password when they first log in. The Passwords entered by users are encrypted using the framework werkzeug.security before being sent to the database. This protects both the user's information and the website's security.

The role-based access controls are what set the users apart. For example, some users may have access to only one of the systems, while only a few may have access to both. Roll call role can access the attendance system but could not access the duty roster system, and vice versa for duty planner role.

Duty Roster System

User Interaction



The Duty roster system shares the same database as the Attendance taking system and filters out a serviceman who are required to perform duties. Duty Roster ICs can use the page above to make any amendments to the serviceman duty type. The system will look through the entire database and only pick out servicemen with the selected duty type. These servicemen will then send their block-off dates before the system start planning their duty roster. In addition, the data includes the serviceman's duty type as well as whether they are working a normal or half-shift shift.

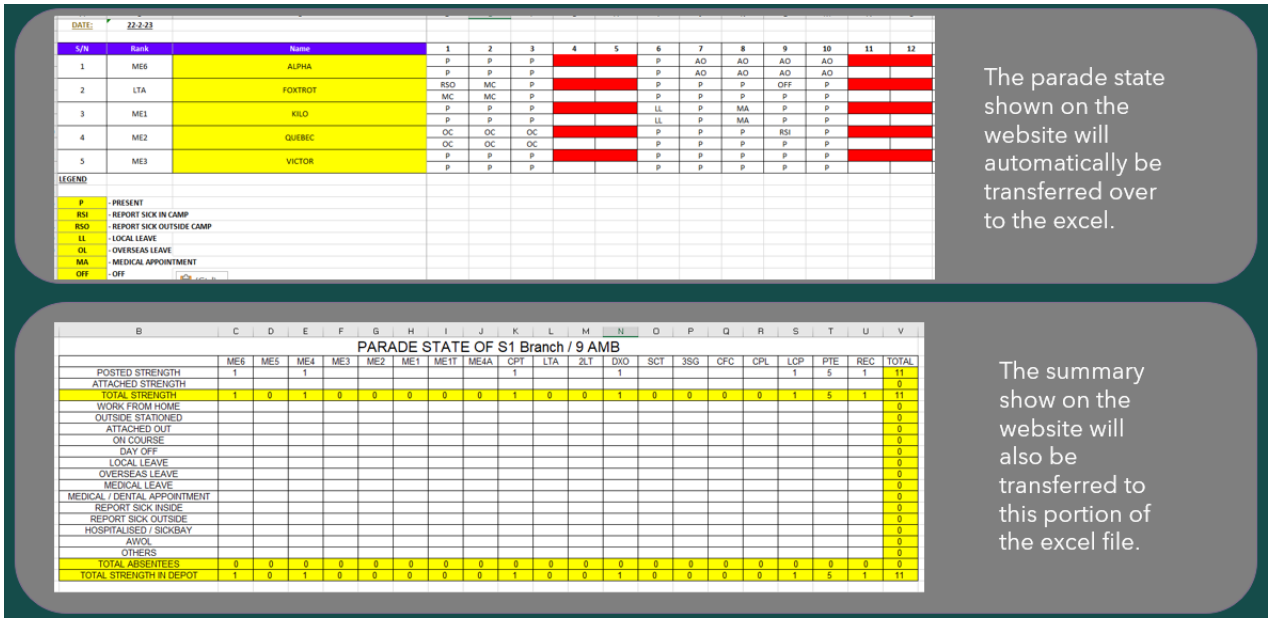
4. Data Transfer to Excel

How it works

The website backend can send data to an Excel sheet using the Openpyxl framework. Because Excel is the only program permitted in the military, the data must be transferred to Excel. Backend functions will include opening the Excel Worksheet and transferring data from the dictionary to the sheet.

Attendance Taking System

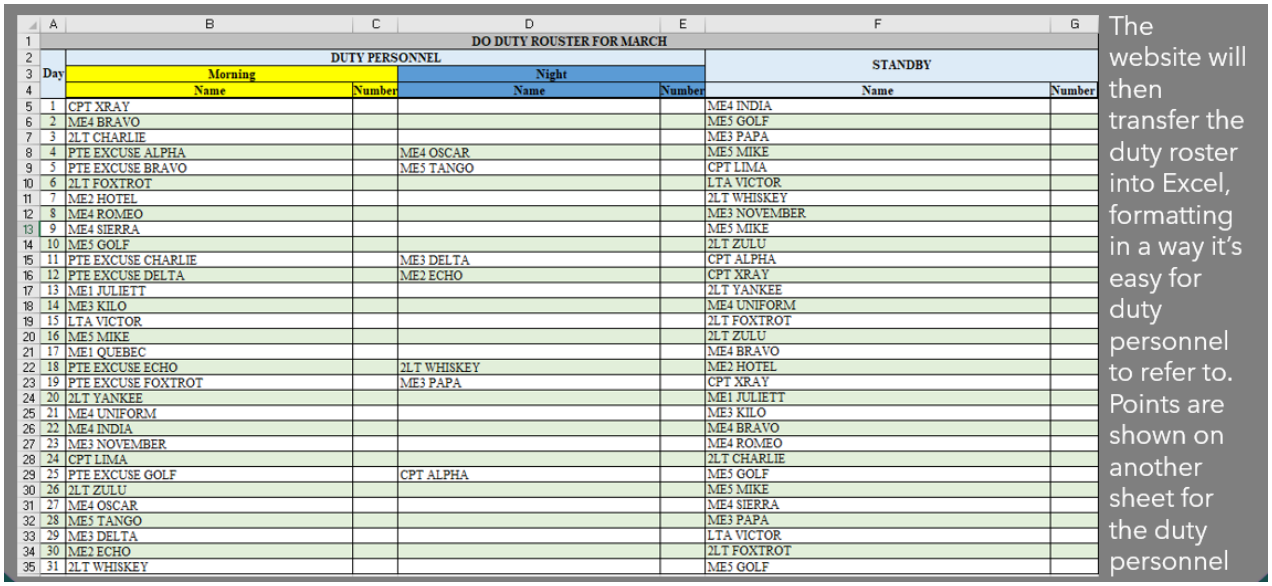
Excel Submission



The table above shows the format provided by the military. Previously, servicemen would take hardcopy attendance and manually enter it into an excel spreadsheet. The attendance-taking system's backend can export all of the data collected on the day to an Excel spreadsheet. The various sheets of the worksheet represent the battalion's various branches and workshops. A loop will iterate through all of the battalion's branches and workshops, transferring attendance and summary information to the sheets.

Duty Roster System

Excel Submission



	A	B	C	D	E	F	G
1	DO DUTY ROUSTER FOR MARCH						
2		DUTY PERSONNEL				STANDBY	
3	Day	Morning		Night			
4		Name	Number	Name	Number	Name	Number
5	1	CPT XRAY				ME4 INDIA	
6	2	ME4 BRAVO				ME5 GOLF	
7	3	2LT CHARLIE				ME3 PAPA	
8	4	PTE EXCUSE ALPHA		ME4 OSCAR		ME5 MIKE	
9	5	PTE EXCUSE BRAVO		ME5 TANGO		CPT LIMA	
10	6	2LT FOXTROT				LTA VICTOR	
11	7	ME2 HOTEL				2LT WHISKEY	
12	8	ME4 ROMEO				ME3 NOVEMBER	
13	9	ME4 SIERRA				ME5 MIKE	
14	10	ME5 GOLF				2LT ZULU	
15	11	PTE EXCUSE CHARLIE		ME3 DELTA		CPT ALPHA	
16	12	PTE EXCUSE DELTA		ME2 ECHO		CPT XRAY	
17	13	ME1 JULIETT				2LT YANKEE	
18	14	ME3 KILO				ME4 UNIFORM	
19	15	LTA VICTOR				2LT FOXTROT	
20	16	ME5 MIKE				2LT ZULU	
21	17	ME1 QUEBEC				ME4 BRAVO	
22	18	PTE EXCUSE ECHO		2LT WHISKEY		ME2 HOTEL	
23	19	PTE EXCUSE FOXTROT		ME3 PAPA		CPT XRAY	
24	20	2LT YANKEE				ME1 JULIETT	
25	21	ME4 UNIFORM				ME3 KILO	
26	22	ME4 INDIA				ME4 BRAVO	
27	23	ME3 NOVEMBER				ME4 ROMEO	
28	24	CPT LIMA				2LT CHARLIE	
29	25	PTE EXCUSE GOLF		CPT ALPHA		ME5 GOLF	
30	26	2LT ZULU				ME5 MIKE	
31	27	ME4 OSCAR				ME4 SIERRA	
32	28	ME5 TANGO				ME3 PAPA	
33	29	ME3 DELTA				LTA VICTOR	
34	30	ME2 ECHO				2LT FOXTROT	
35	31	2LT WHISKEY				ME5 GOLF	

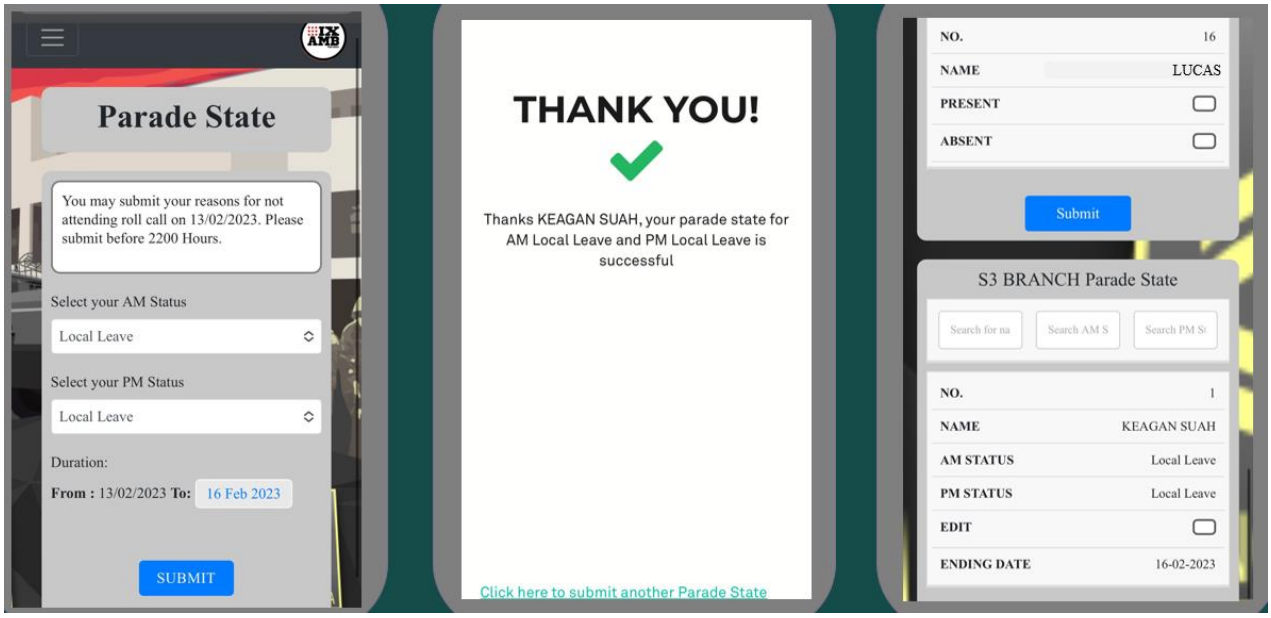
The website will then transfer the duty roster into Excel, formatting in a way it's easy for duty personnel to refer to. Points are shown on another sheet for the duty personnel

The table above shows the template for how servicemen will view their duty roster and the type of duty they will perform. Once the Duty Roster System algorithm has generated the duty roster, it will be transferred to the Excel Worksheet. The Openpyxl framework can also read the Worksheet, receive its data, and send it to the website's backend. This is necessary because it makes duty management easier. The title cell will also automatically change depending on the date the machine schedules the duty.

5. Automated Attendance Taking

Individual Submission

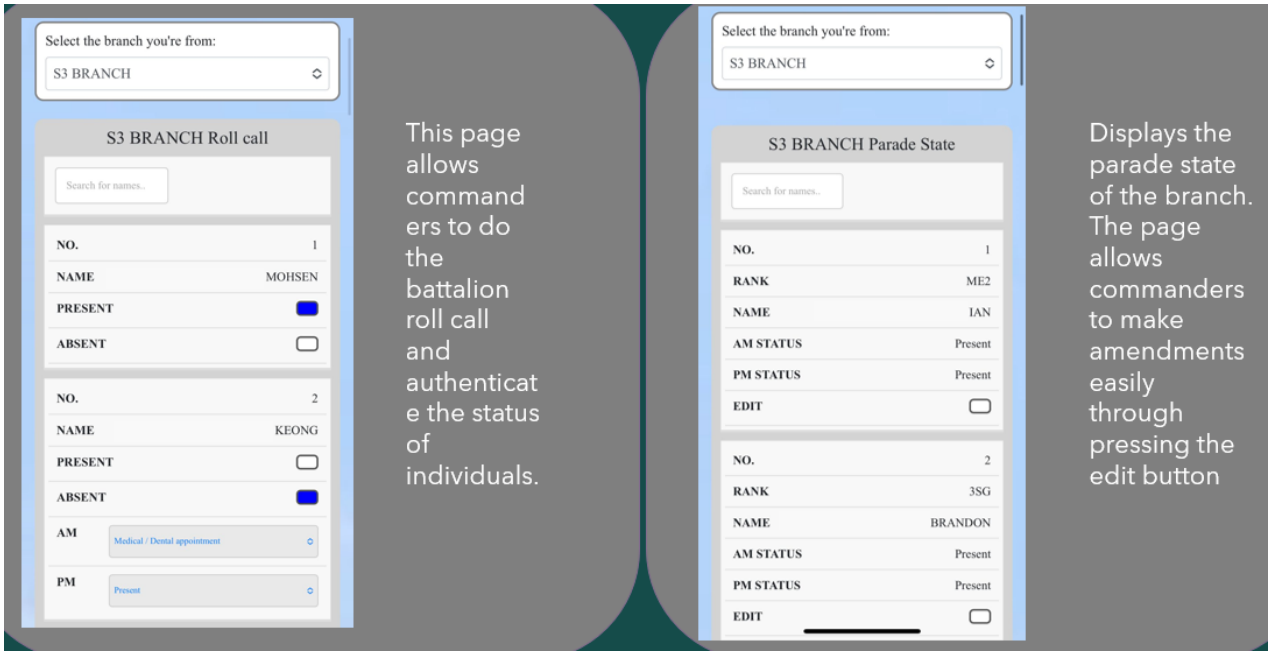
User Interaction



The page shown above is where service members will report their attendance status. The page to the left, allows all servicemen to submit their attendance status if they will not be present for roll call the next day. Servicemen can specify when their status will end, and the website will automatically submit for these servicemen until the end of the duration. The data of these servicemen will be transferred to a dictionary, which will store their attendance status and duration. All of the dictionary's data will be transferred to the current attendance dictionary, which is used to display attendance.

Commander’s Attendance Taking

User Interaction



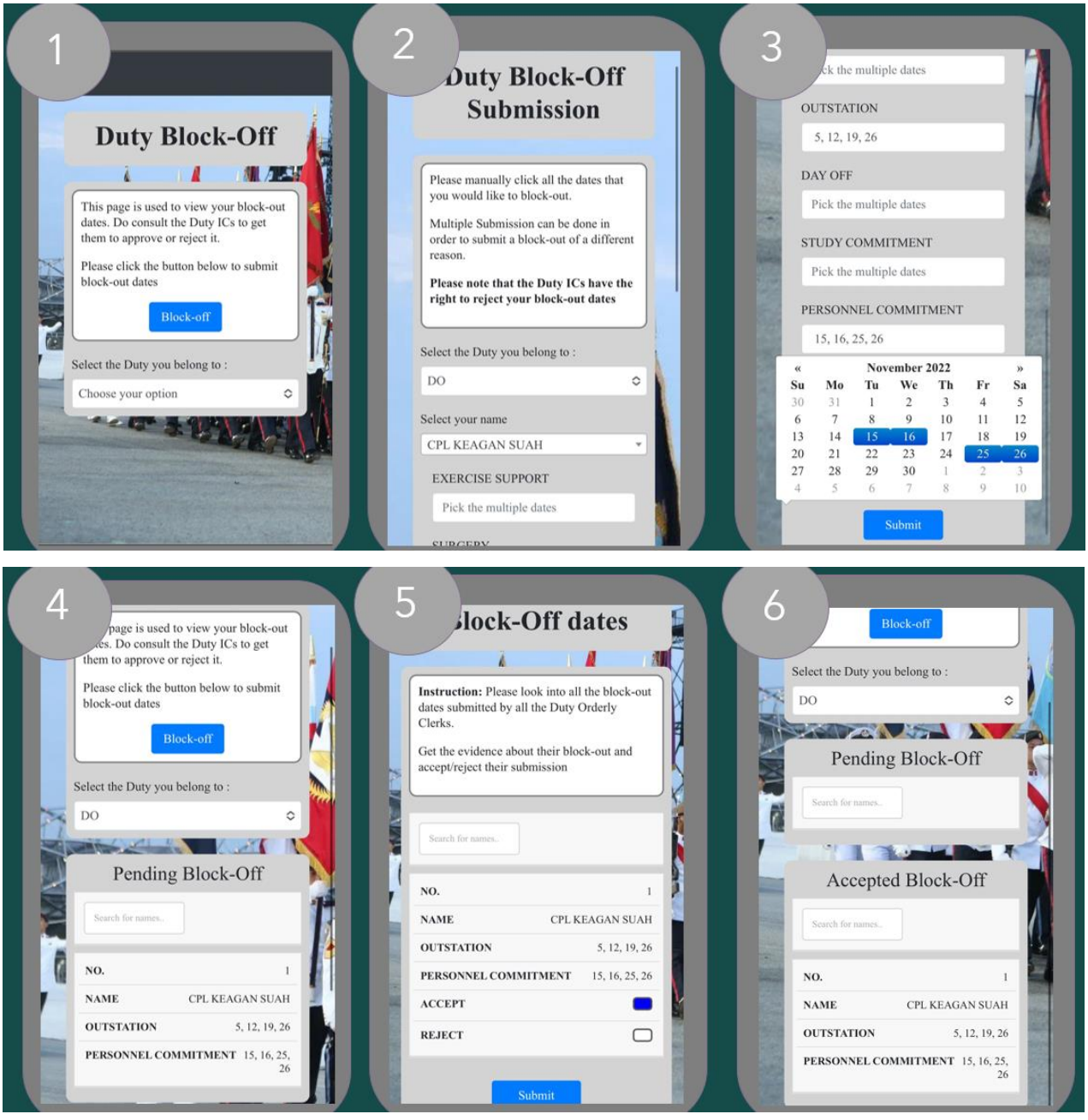
The roll call list and the finalist attendance list will be displayed on the page shown above. Serviceman who has a roll call role can access these pages. All data, including servicemen and their ranks, will be collected and entered into the roll call dictionary by the Attendance Taking System. Marking the servicemen as present indicates that they are present in both the morning and afternoon, whereas marking absent necessitates the roll call in charge to enter their status in both the morning and afternoon.

When the roll call in charge marks a serviceman as present or absent, the serviceman is removed from the roll call dictionary and transferred to the current attendance dictionary. The frontend will be updated, and the changes will be visible. Once all of the names on the roll call list have been marked, the dictionary will be emptied. The frontend hides the roll call table when the roll call dictionary is empty, displaying only the current finalist attendance list.

6. Automated Duty Roster

Individual Submission

User Interaction



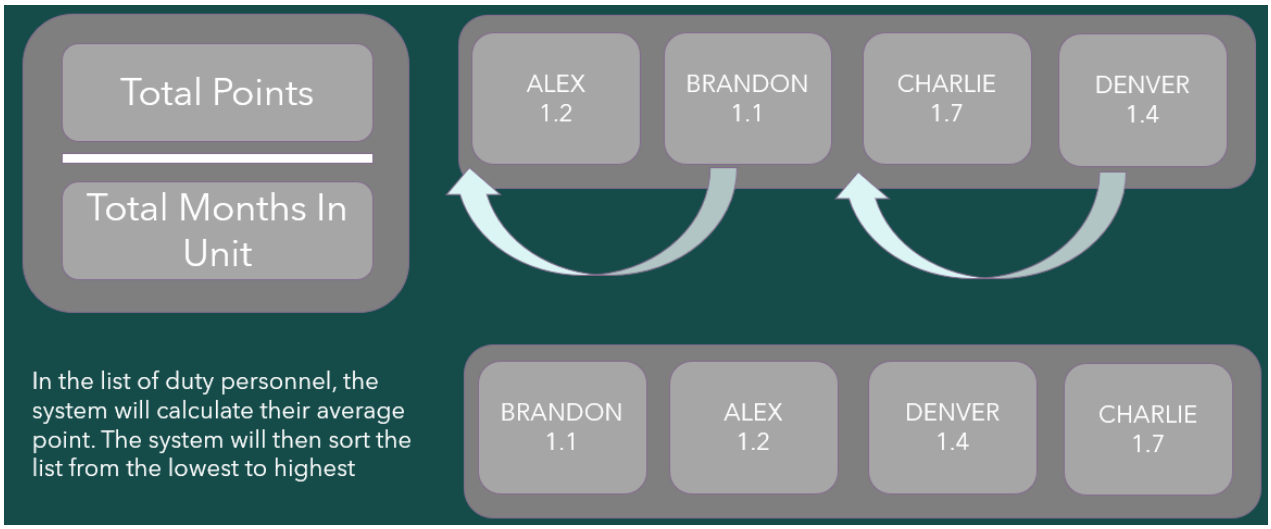
The image above depicts the steps and processes that servicemen must go through to submit their block-off dates. Block-off dates are times when a serviceman is unable to

carry out his or her duty. Servicemen must submit their reason for their inability to perform the duty and their respective dates. Two dictionaries are among these features: the pending dictionary and the accepted dictionary. Servicemen will enter particulars along with reasons for their inability to perform duty.

The collected data will be transferred to the pending dictionary, which the serviceman will be able to view on a block-off page. Serviceman with Duty Roster role will have administrative access to the Duty Roster System and will accept any data entered into the pending dictionary. Accepted submissions' data will be transferred to the accepted dictionary, while rejected submissions' data will be removed from all lists.

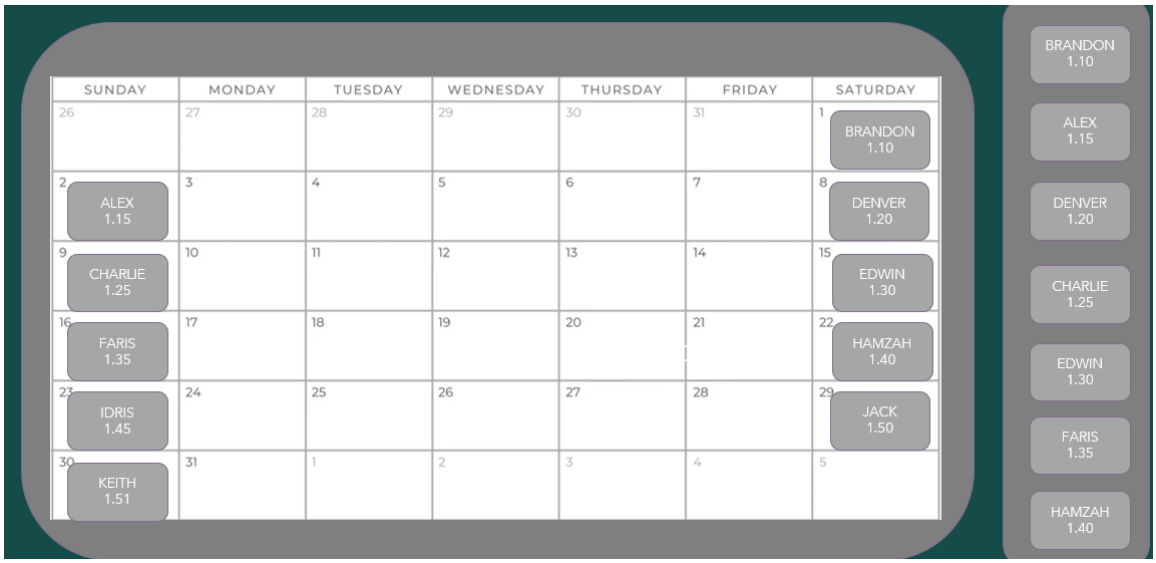
Duty Roster Algorithm

Algorithm Explanation



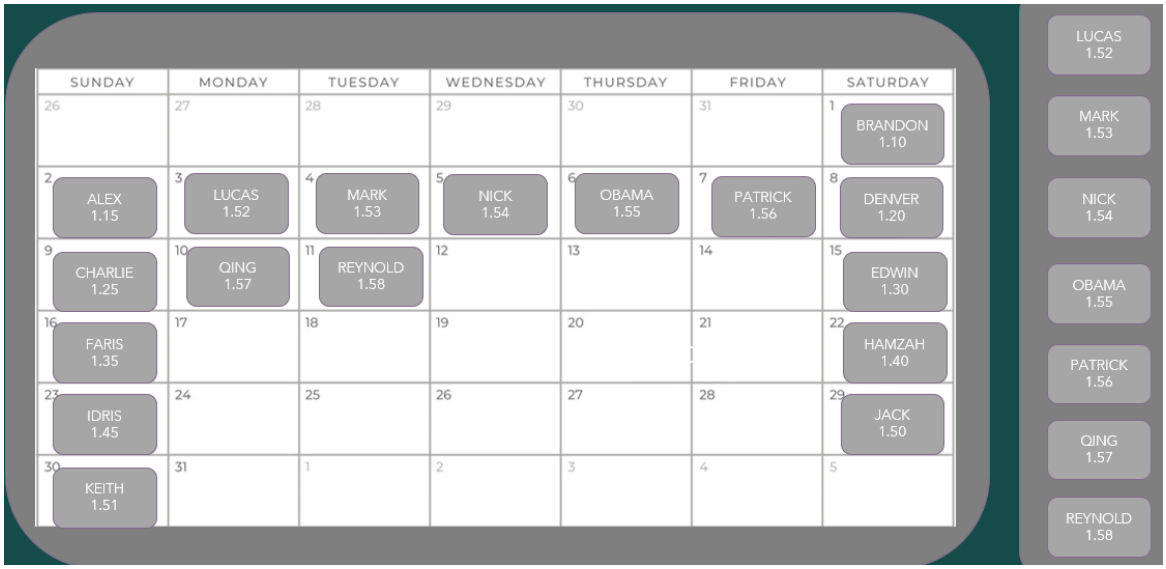
The Duty Roster System's main feature is that it eliminates human errors and reduces the time required for manual planning, which takes 5 days, to seconds. All servicemen attributes, such as points, months in the battalion, and block-off dates, will be retrieved by the Duty Roster System and stored in the name list dictionary. The average point for each serviceman in the name list dictionary will be calculated by taking their total points and dividing them by the number of months they've been in the unit. The dictionary will be sorted based on the average point value after the calculation. The dictionary will prioritize low-average pointers.

Algorithm Explanation



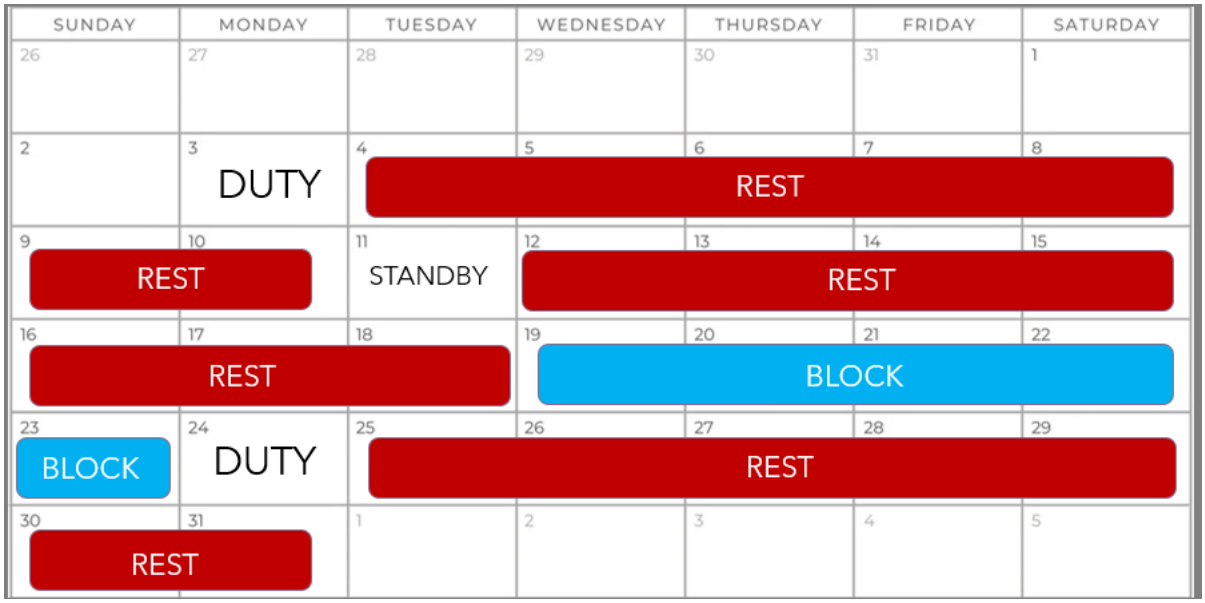
The system's goal is to have a low standard deviation for all servicemen in the duty database. This is done to ensure that duty planning is fair and impartial. To plan the duty from the beginning to the end of the month, the system will first retrieve the sorted name list dictionary. The Datetime framework can be used to determine whether the day is a weekday or a weekend, which will be useful later in the planning process. The Duty Roster System will run two loops to ensure fair planning.

Algorithm Explanation



Weekends have more points than weekdays, the first loop will ensure that low pointers are scheduled for duty on weekends. The serviceman that is scheduled will be moved to the back of the dictionary, the cycle will repeat until the loop is completed. Following the completion of the loop, the System will invoke the functions that will update the average points and sort the name list dictionary once more. The System will then execute the second loop and schedule the remaining weekdays based on the logic of the first loop.

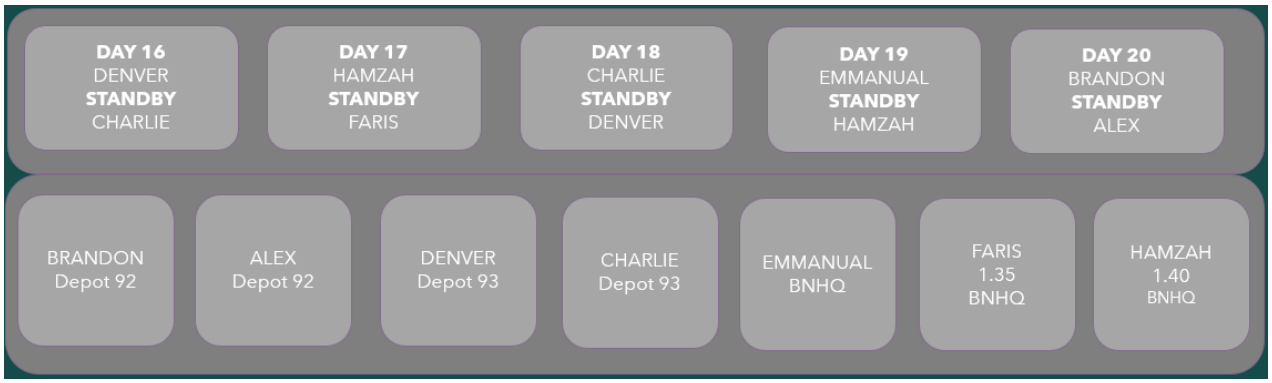
Algorithm Explanation



The System must follow these criteria when planning the aforementioned duty. The first requirement is that servicemen rest for at least seven days before or after their scheduled duty. The system does this by subtracting an individual scheduled duty day from the current date the system intends to schedule and scheduling servicemen only if the result is greater than 7. The system will then look for the next lowest serviceman in the dictionary until a person is scheduled.

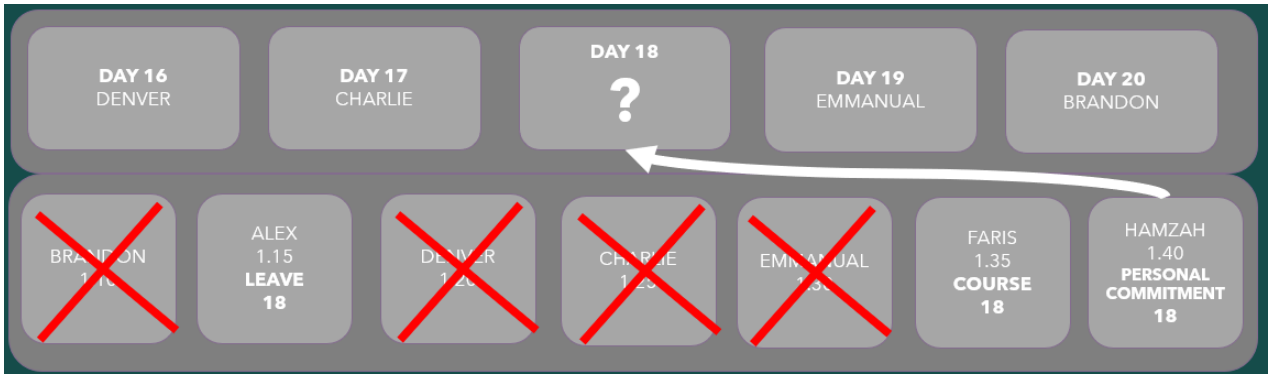
The second requirement is that the scheduled duty does not conflict with a serviceman's block-off date. The System does this by aggregating all of the block-off dates and producing a list of dates. The System will check to see if the scheduled day is not on the list of block-off dates before scheduling a serviceman. If that is the case, the system will move on to the next serviceman.

Algorithm Explanation



Due to the uncertainty surrounding the pandemic, standby is required if the duty serviceman tests positive for COVID-19. Based on the two criteria mentioned above, the system must schedule the standby serviceman. Furthermore, when scheduling standby, two additional criteria must be met. The first criterion is that the standby date cannot be less than seven days before the actual duty date. This function is carried out by the System in the same way as the 7-day rest function. The second requirement is that the standby personnel be from the same branch or workshop as the duty personnel. This is because servicemen from the same branch or workshop are less likely to sabotage each other.

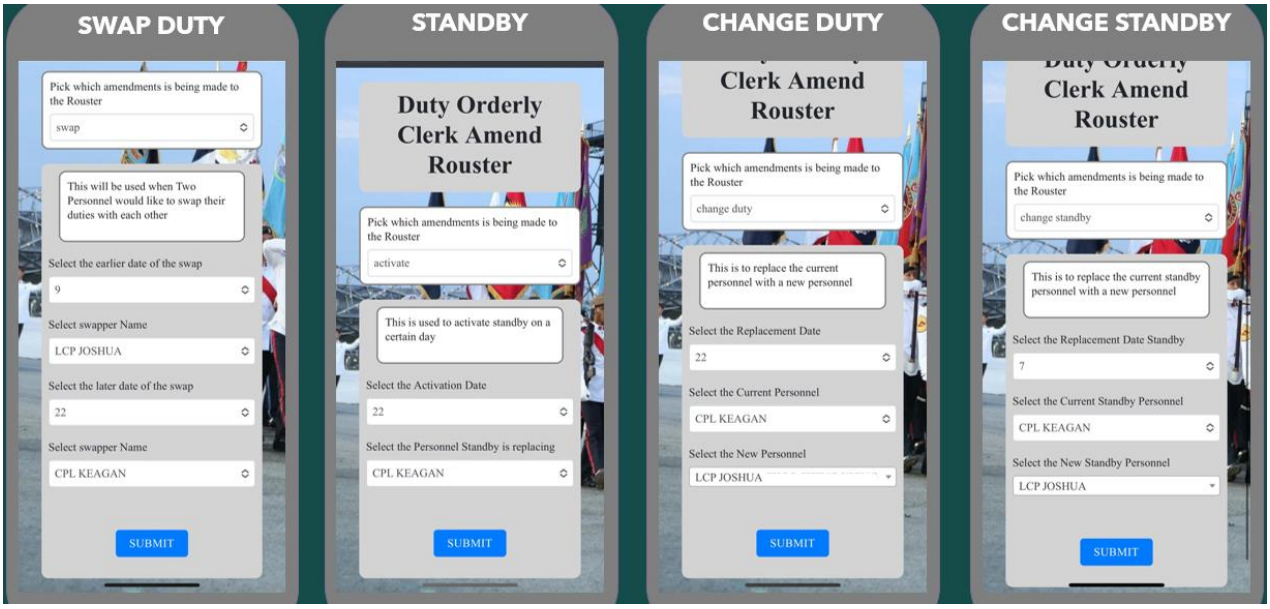
Algorithm Explanation



There may be days when the system is unable to schedule a serviceman. The system will remove some commitments, such as personnel and study commitments, and reschedule using the list. If no standby days are available, the standby criteria at the same depot will not be applied. This is accomplished by the System looping through the name list dictionary and removing one of the lowest priority block-off reasons. As a result, the total number of blocked-off dates will be reduced by one.

Managing Duty

User Interaction



The features shown in the picture above aid in the management of duty rosters. Only Duty Roster role have access to this page. A service member may wish to swap or change their duty or standby status. Managers or ICs must make the necessary changes and assign the appropriate points to the days the serviceman is on duty. Using the page shown above, users can make changes to the roster generated by the System.

A function is written in the Openpyxl framework to read and extract all of the data from the excel sheet and save it to an amending dictionary. The frontend will then display the dictionary for the user to edit as needed. Following the changes, the dictionary will be updated, with the new keys and values overwriting the old, and the dictionary data will be returned to the Excel sheet. The system will also retrieve data for the names and total points of the selected servicemen. The system subtracts the serviceman's previous points and adds the correct number of points earned.

7. Conclusion

The Automated Attendance Taking System makes it easier for servicemen to manage their attendance. The main problem with the traditional method was that it required a military computer to submit the attendance. The issue is that the majority of the servicemen are at their workshop and do not have access to a military computer. This Automated Attendance System creates a platform for service members to manage their parade status on their phones. The system contributes to reducing the total number of hours required to complete the battalion parade state.

The Automated Duty Roster System provides a platform for duty management as well as automatic duty planning based on logic. Traditional duty planning used to take a long time to plan because the duty roster had to follow the complicated logic of duty planning. Human error is also possible, which is unsettling in a military setting. Because it follows the duty planning logic, the System provides unbiased duty planning. Aside from duty planning, duty managing is an important attribute because many changes are made throughout the month, so assigning the appropriate number of points and rest is necessary to ensure fair future planning.

Future Enhancement

After seeing the website and the two automated systems, the battalion's commanding officer was impressed and suggested two improvements. The first improvement concerned the Automated Attendance Taking System. Commanding Officers want workshops to be able to download their parade state, which includes a summary of their entire depot.

The Automated Duty Roster System was the subject of the second enhancement improvement. Allowing duty servicemen to choose their preferred date to perform their duties will motivate the battalion to perform better. Based on the feedback I received, I need to modify the duty roster algorithm by including options for duty servicemen to indicate their preferred duty. The servicemen will be sorted from highest to lowest points, with higher points having a better chance of getting their preferred date.

Testimonial Received

LETTER OF RECOMMÉNDATION

I am writing to recommend **Mr Keagan Suah T0135163F** for admission into his chosen university or future occupations. Keagan has been serving his Full Time National Service in one of my departments for the past year and has demonstrated his abilities in the areas of IT.

Keagan has been a valuable asset to the battalion. His superiors are comfortable in letting him plan and execute tasks independently, knowing that they would be completed meticulously, timely and accurately. In addition, he is flexible and adapts quickly to a wide variety of tasks that the department undertakes, another plus point for us to have him in the team.

Keagan is a firm believer of continuous self-improvement. With a keen interest in IT and programming, he had been attending relevant online courses after work in effort to strengthen his expertise in programming. In addition, Keagan understands that with programming, 'practice makes perfect'. Keagan has put his skills to good use - he identified two problem statements and developed IT solutions that made our administrative tasks more efficient and effective:

a) The Battalion Parade State System digitalises the battalion's personnel attendance capturing, verification and compilation process, improving efficiency and reducing errors. Keagan uses Python, HTML, CSS, and JavaScript to develop the user interface, server, application and database of the progressive web app. Adopting an agile development approach, he was able to work closely with all stakeholders to deliver a complete product, incorporating users' feedbacks, within 3 months whilst juggling daily work requirements.

b) The Automated Duty Planner digitises and automates the battalion's duty personnel roster planning process. Keagan developed the algorithm that incorporated the rules and considerations to automate the process of duty planning. This reduced the manual duty planning process by close to 99%.

I believe Keagan processes the gumption, diligence and interpersonal skills that will allow him to do well in any endeavor he embarks on. His keen interest in IT and his eagerness for self-improvement will certainly allow him to excel in the above mentioned course he is applying for with SIT. I am certain that Keagan will be a valuable asset and an active contributor to the course to aid not just his own learning, but the learning of his peers.

If you have further queries about the above-mentioned and / or Keagan's character reference in general, do feel free to reach me through email at **Tay_Wei_Jie1@defence.gov.sg**

Best Regards,
Military Expert 6 Tay Wei Jie
Commanding Officer
9th Army Maintenance Base