# **SimplifyNext RPA Hackathon 2023**

**Title: Chair Share**

# 

# **Members:**

|  |  |
| --- | --- |
| **Name** | **Email** |
| Hendy | hendylim74@gmail.com |
| Kwek Ee Chern | KWEK0076@e.ntu.edu.sg |
| Lim Wei Yang | WLIM125@e.ntu.edu.sg |

Libraries play an important role in providing a common space for students and members of the public to gain access to educational materials. However, with the escalating demand for library seating, it has become increasingly imperative to develop a streamlined and user-centric system to handle bookings, particularly for groups. This is especially critical for students who seek to work on collaborative projects or study in groups. Regrettably, the current system for booking library seats for groups is insufficient and has resulted in dissatisfaction among its target users. Thus, the objective of this statement is to scrutinize the deficiencies of the current system and present viable remedies that enhance the group booking experience for library patrons.

The group came up with a solution to alleviate the frustration from booking seats at a library. Our UIPath application aims to assist in finding seats in a nearby library and facilitate the process of booking multiple seats in a library. By using UIPath, we have implemented a robot that will scrape through the NLB library book site to identify which libraries have seats available and book seats in the same area for students in a group.

Prerequisite

* Google Chrome
* Set permission for UiPath Web Automation Extension to work in incognito mode
* User must have access to the folder in the cloud to get assets (NLB usernames and passwords are stored on the Uipath Cloud).
* Main file is **Final\_Flow.**

Flowchart based on the Workflow

|  |
| --- |
|  |

Figure 1

The workflow is splitted to 3 different pipelines namely the initial,inner and final pipelines. First in the initial pipeline, users can input their selections and based on the inputs, users’ information is retrieved from the cloud. Secondly, we have the inner pipeline where the bulk of the algorithm can be found. Its main purpose is to perform the actual query process and place the booking for the group of users. Lastly, we have the outer pipeline where the booking has already been done and visual feedback in the form of telegram messages will be sent to the user to notify them on their booking details and whether it is successful or not.

Pipelines

Right before we get into explaining the various sectors in the pipeline in detail, the list of arguments that we used can be found in the table below.

List of Variables/Arguments

|  |  |  |
| --- | --- | --- |
| Arguments | Type | Purpose |
| Start\_time | String | Contains the user’s booking time |
| Duration | String | Contains the user’s duration of stay in the library |
| SelectLib | String Array | Contains a list of user selected libraries |
| Total | Integer | Number of selected users as indicated on the form |
| C\_AVAIL | Boolean | Determines whether the algorithm will put a filter in place when querying the list of libraries in the table |
| Found | Boolean | Indicates whether suitable seats based on the user’s choice were returned from the library table |
| SEAT\_TO\_ORDER | DataTable | Contains multiple arguments used for the seat booking algorithm |
| SELECTED\_USER\_DT | DataTable | Contains selected user’s details to be used to pull credentials from orchestrator |

Initial Pipeline

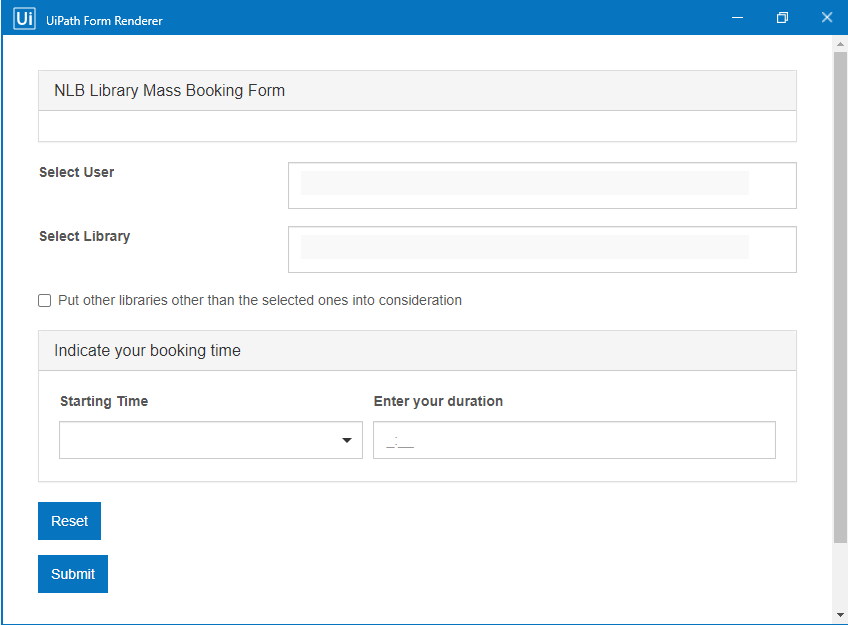


Figure 2

When the user first runs the robot, the robot will first need to get the details to display as options in the input form as shown above. This is done by using the Uipath Orchestrator API where user credentials are saved under assets which can be accessed with the use of the api. Other input options such as the library field and starting time field are dynamically generated based on the parameters such as the opening hours of the library and the current time when the user runs the robot.   
  
After the user has successfully filled up the form, the options are then parsed into a SEAT\_TO\_ORDER, a datatable which is then passed to the inner pipeline for the booking algorithm to work.

Inner Pipeline

A found variable will be initiated as false, It wil be an indicator that suitable seat has not been found. When the program started for the very first time, the system will get Library data shown below (Same Tableid indicated the seats are in the same table).

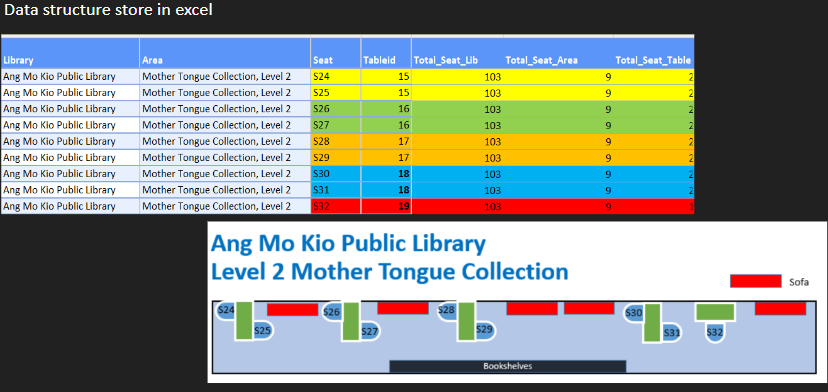


Figure 3

The system will then login to the NLB website to collect the real time update of the seat available.



Figure 4

The following is algorithm to check the seat available by extracting it as a string (1 and 2 in Figure 4) and compare it with the available data from excel (3 of Figure 4) It will filter out all the seat that is not available and count the number of seat (4 of Figure 4). Finally filter out if the count is lesser than group size and sort the count from smallest to biggest (5 of Figure 4). This way we can obtain the most suitable seats for the group.

In inner pipeline, if the seats are found, it will be allocate to all the users in SEAT\_TO\_ORDER and variable found will be set to true so that the next iteration the inner pipeline will proceed with booking a seat for the users.

Output

For our output, we used a telegram bot through the pyTelegramBotAPI library to inform our users of successful bookings.

The python script is being run from pythonanywhere.com and is being run as a webapp through the use of Flask. By implementing it as a webapp, the end user does not have to install python and its dependencies, and the bot’s API key can be protected on the server. Furthermore, it is essential as a dictionary of username:user-id data has to be maintained by the python script.

The python script takes in a receiver’s username and message value into the url and sends it through the send message function by referencing the dictionary of username:user-id.

Lastly, the robot enters the webapp like in the sample url below:

<http://hackathonserver.pythonanywhere.com/send?receiver=pistato&message=hello+world>

Going to this webpage will send a message to Pistato with the text, “Hello World”. However, due to limitations imposed by Telegram, the message can only be sent if the user has used the /start command with the Telegram bot at <https://t.me/RPAHackathonNewBieBot>

Thus using the get request activity on UIPath, we send a request to the webapp and in doing so, send a message to the telegram user.

Workflow not includes

* Error handling if the user already has a booking.

(NLB only allow one booking per user)

* If the booking duration exceeds the end time allocated for online booking
* This solution does not include datetime for user input because the booking system only allow on the day and the day after booking.