Childcare-Parent Liaison Aide: Project Report

Submitted in partial fulfillment of ISY5005 Intelligent Software Agents:
Practice Module 1

Team Members:

- ANG PAU HUANG, EDWIN/ A0195275U
- CHEN MINGYI EDMUND/ A0031105E
- CHEOK MEI LI/ A0165420N
- PADMAPRIYA MATHIVANAN/ A0215281M

NUS National University of Singapore | **ISS** INSTITUTE OF SYSTEMS SCIENCE

Contents

1 Executive Summary

Liaison work between a service provider and a consumer is a common task that is primarily undertaken by a human worker. However, our team saw the potential for RPA to aide in a human liaison officer so long as we could capture the human thought process and the workflow. We set our eyes on the liaison tasks in childcare centers, because as parents of toddlers ourselves, we had experienced first-hand the amount of work that the center needed to go through with anxious parents, many of whom, had many queries for the center.

The job scope of a liaison officer in a childcare center is wide but for project purpose we had scoped it to address the repetitive, ad-hoc, and mundane tasks involved in the open-house invitation. Our team created two bots to aid the human worker.

An email bot was chosen as we felt invitations sent through email held a more official and professional outlook compared to, say, text messages. This software robot handled the tasks of sending out email invites informing parents of the month's open house dates, gathering parent's availabilities on the dates, informing parents of their allocated open house slot, and sending out appropriate reminder emails. The email bot was also designed to attend to ad-hoc requests (e.g. change requests for allocated slots) however we felt that ad-hoc communication through emails was not common, instead a messaging app was more appropriate due to it greater accessibility.

A chatbot was thus created to handle ad-hoc requests, as it was, in general, more natural for an individual to turn to messaging apps for day-to-day ad-hoc communications. The tasks fulfilled by the chatbot included: editing availabilities, handling requests for change of allocated open-house slots, handling enquiries regarding allocated slot or availability, and other basic general enquiries.

To realize our ideas, we had used a variety of tools, and packages. UiPath was used to put together the email bot, while python + DialogFlow + Telegram formed our chatbot. With these, our bots could aid in the end-to-end task of liaising with parents from invitation till their attendance on actual event day.

2 Background

2.1 Problem Description

Childcare – Parent liaison is a complicated process involving numerous back-and-forth between the center and the parent. This is still the case even for basic liaison task such as Open House invitation. Figure 2 concisely summarized the sequence of actions required by the center for end-to-end open house liaison work. We could visualize the elaborate workflow, and the multiple actions and decision points required by the center. From the workflow chart, we identified the tasks/ actions appropriate for RPA.

2.2 Market Research

From our own experiences interacting with childcare centers, we suspected none of the centers we communicated with had extensive, or in fact, any, automated software agents aiding in the liaison process. Telltale signs of this included the slow response from centers. We figured that a recurring problem among most childcare center operators would be staff resources spent in liaising with parents.

Compounding on this problem was the expected growth in childcare centers as reported[1] in Asiaone's article; which would lead to more human resources needing to be hired to fulfill the role of childcare-parent liaison officer.

Hence it struck our team that creating an RPA solution to aid in liaison task was solving a real ongoing problem which had increasing market demand.

Preschool Enrolment Statistics Singapore 2019

Year	2012	2013	2014	2015	2016	2017	2018	April 2019
Total no. of child care centres	1,016	1,083	1,143	1,256	1,342	1,419	1,495	1,517
Total no. of child care centre places	92,779	101,597	109,694	123,327	137,278	149,803	167,421	171,660

Figure 1: No. of centers are growing year-on-year, resulting in growing demand for automation solutions.

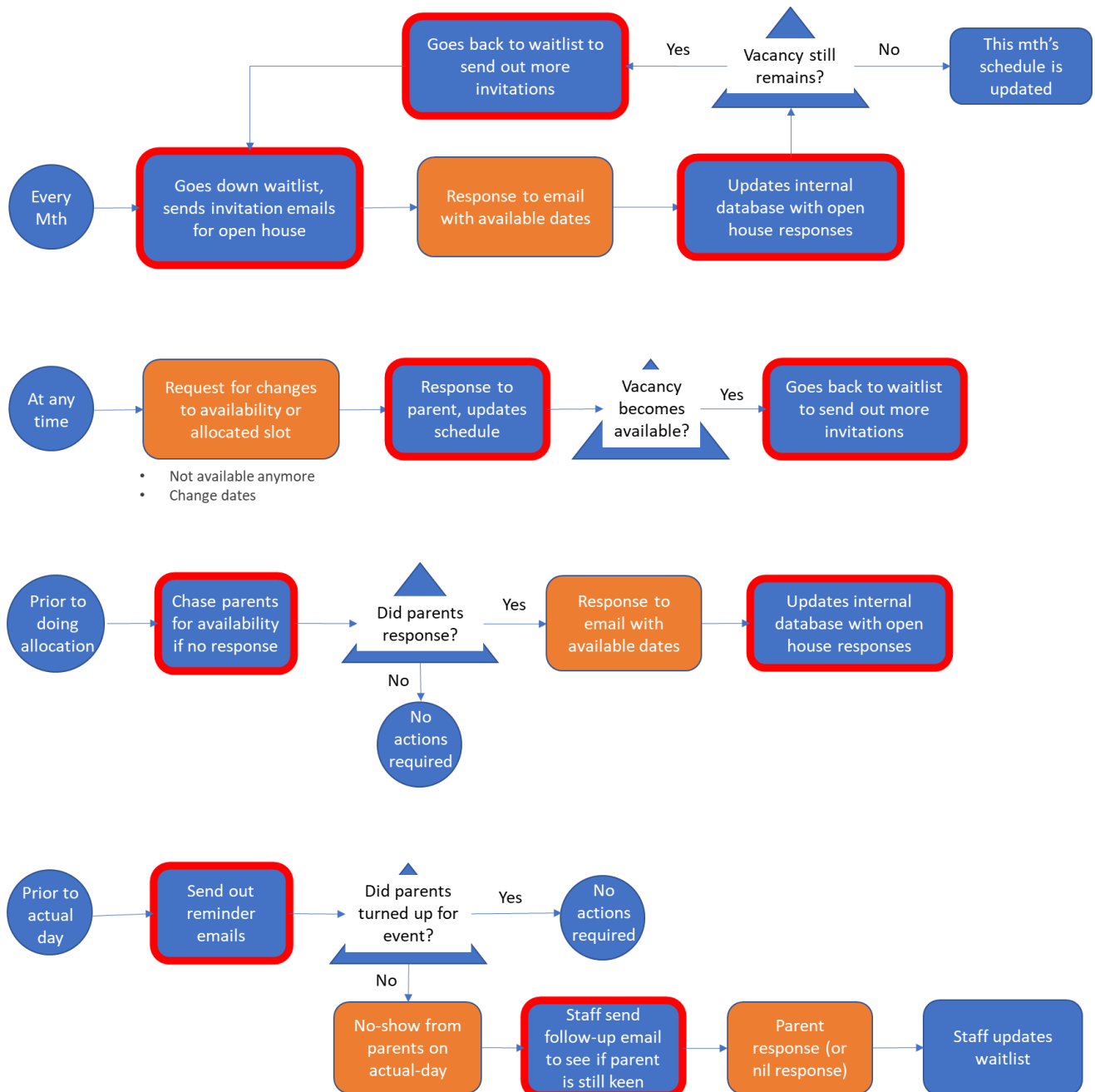[1] https://www.asiaone.com/singapore/why-childcare-supply-and-demand-dont-add

Figure 2: Workflow for Open-house Liaison

3 Objectives and Success Measures

In a motherhood statement, our project objective is to deploy a working MVP to aid in childcare-parent liaison but specifically for the process of Open House invitations. With the help of software robots, we hope to offload as many tasks as possible from the human worker.

In more specific terms, we had understood and mapped the end-to-end workflow from sending out the first invitation till a parent's attendance on actual Open House day into a flow chart shown in Figure 2. We then identified the tasks within the flow chart that had potential for RPA to intervene. A measurable deliverable for this project was, thus, the successful automation of each of these tasks identified (boxes with red outline in Figure 2).

S/n	Task
1.	Sending out of Open House Invitation to gather parent's availability for list of dates
2.	Collection of parents' replies in availability dates
3.	Sending reminder emails to chase parents for availability dates
4.	Auto-allocation of parent to an Open House slot
5.	Attending to changes in availability raised by parents
6.	Attending to parent's requests for changes to allocated Open House slot
7.	Attending to enquires on allocated Open House slot
8.	Attending to enquires on the submitted availability
9.	Sending reminder emails to parents prior to their allocated Open House slot
10.	Sending follow-up emails after parent's attendance to an Open House event
11.	Attending to general enquires regarding center information

Table 1: Tasks identified for RPA

The success of our product would be measure by how successful we were in offloading the tasks specified in Table 1 from a human worker.

4 System Design

4.1 Broad System Design

Our system consisted of two major parts, an email bot and a chatbot. The email bot was primarily used to initiate conversation with the parent at specific junctures in the whole workflow. Its tasks include

1. Notifying parents of open house dates at the start of every month,
2. Retrieving replying emails from parent in the middle of every month

3. Informing parents of that their allocated Open House date
4. Sending reminders on the allocated Open House date closer to date
5. Sending follow-up emails after parent's attendance to an Open House event

The chatbot, on the other hand, was used to attend to ad-hoc enquires or requests from the parent. It was designed to:

1. Collect parents' availability dates
2. Attend & process changes in availability raised by parents
3. Attend & process parent's requests for changes to allocated Open House slot
4. Attend to enquires on allocated Open House slot and the submitted availability
5. Attend to general enquires regarding center information

Figure 3 shows the broad system design, the email bot and chatbot performed their designed tasks independently, but ultimately their outputs were synchronized at a centralized shared database.

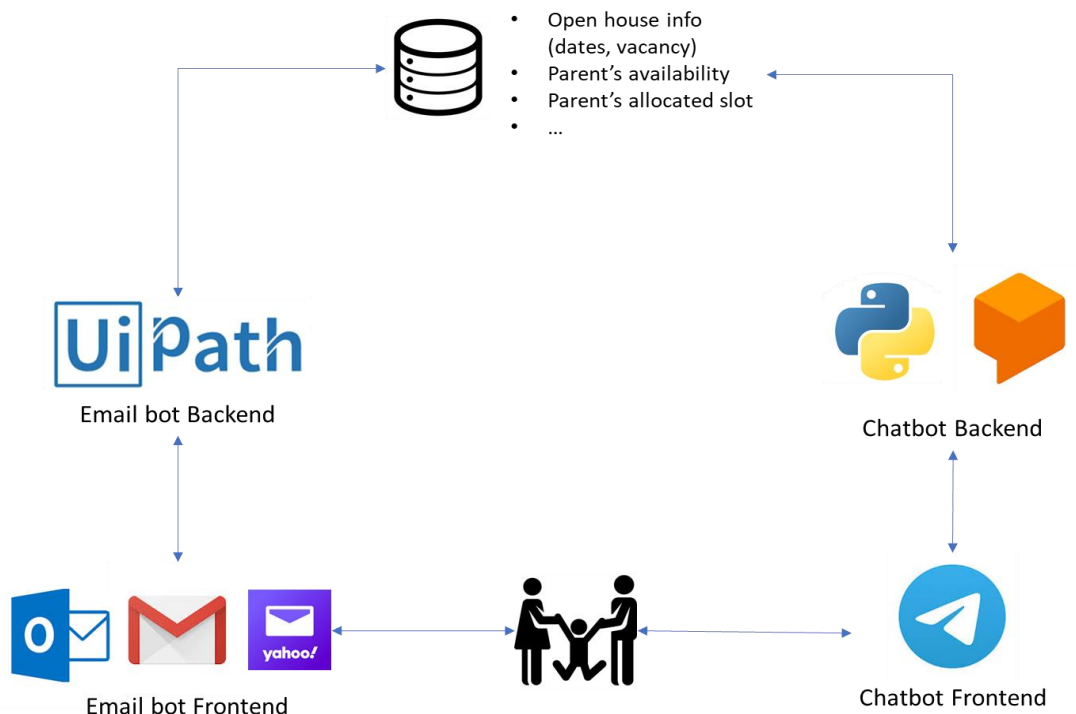In section 5, we elaborate in detail how each bot was implemented and the internal workings of each.

- Open house info (dates, vacancy)
- Parent's availability
- Parent's allocated slot
- …

Email bot Backend

Chatbot Backend

Email bot Frontend

Chatbot Frontend

Figure 3: System Design Overview

5 System Implementation

5.1 Email Bot

<u>Figure 4: System Design of email bot</u>

The email bot interacts with the Childcare Center's Outlook app, sending and receiving emails through this app. It supported the human worker on the following tasks (which we will elaborate under subsections of 5.1):

1. Sending Open House Invitation
2. Email to Inform Parents on Open House Vacancies
3. Email Reminder on Open House Appointments
4. Email Follow-up with Parents on Missed Open House Event
5. Check Parents' Response, Update Waitlist, Obtain Insights

We chose Outlook because it is a popular email app amongst companies. According to Gartner's 2019 market research, Microsoft had 87.5% of the email and authoring market[2]. In addition, one in five corporate employees now use an Office 365 cloud service[3].

However, this does not mean that the parents must send email from Outlook platform. Our email bot can retrieve emails from various other platforms (e.g. Gmail, Yahoo, etc) through the Outlook app. As the email bot needs to interact with Outlook and Excel database, we chose UiPath since it has ready activities interacting with various Microsoft Office products.

---

[2] IMicrosoft created the office suite status quo. Can Google grow? - https://www.ciodive.com/news/Google-Microsoft-Office-collaboration/571740/

[3] Microsoft Office 365 is Being Adopted and Used at an Enormous Rate - https://blog.goptg.com/microsoft-office-365-statistics

5.1.1 Sending Open House Invitation, Processing Parents' Replies

The first interaction with the parents will be a templated email (see <u>Figure 5</u>) sent by the email bot. These are the checks that are done before the email is sent:

- ✓ We have at least 3 open house dates with vacancies, to ensure that parents are given flexibility to choose the dates.
- ✓ Parents who are looking to enroll their children sooner will get priority on the waiting list
- ✓ The total number of parents we send emails to is equal to 120% of the number of total vacancies we have across all available open house dates.

Next, the email bot will check for replies from parents after a suitable interval. <u>Figure 6</u> shows the steps the bot goes through before sending an email with the confirmed Open House date to the parent.
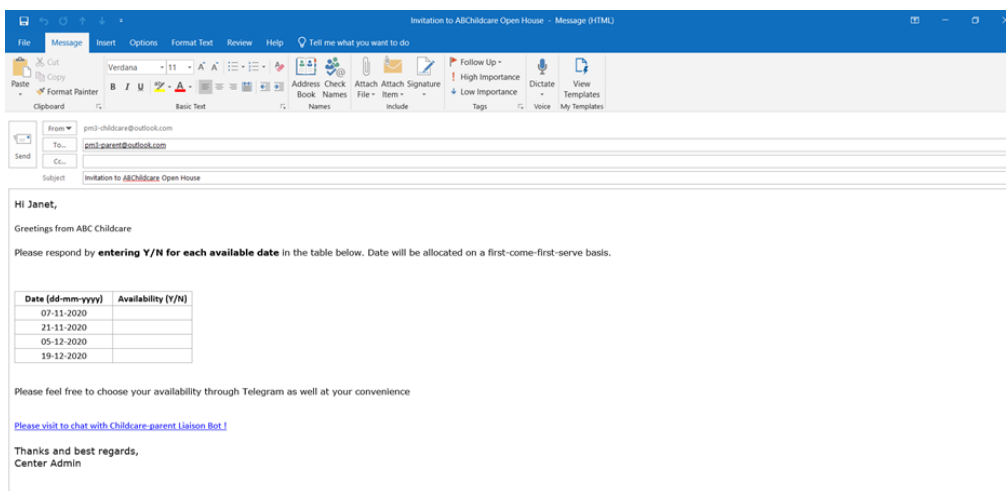

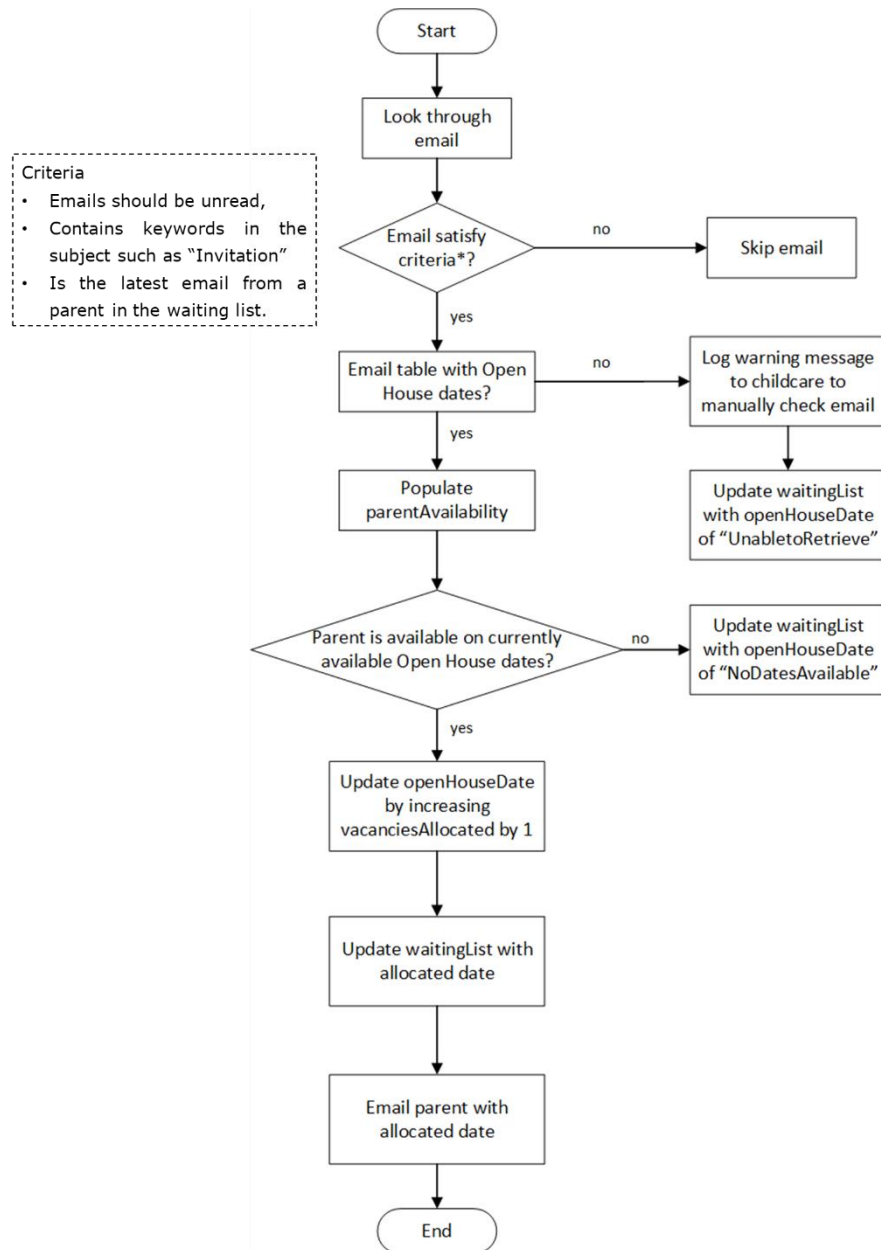Figure 5: Email template for childcare to parents

Figure 6: UiPath flow chart for receiving replies from parents

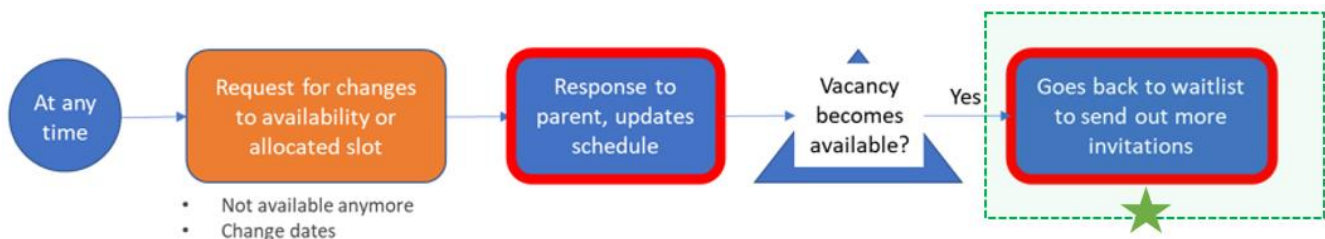## 5.1.2 Email to Inform Parents on Open House Vacancies

Figure 7: Workflow for Human Liaison

Parents can have preferences for open house dates which were not available at the point in time of their request. Due to cancellations of by other parents, it is possible that vacancies for their requested dates can become available at a later date.

There are repetitive tasks today where a human childcare center liaison worker today will:

- Go through data tables (from database) containing information about vacancies row by row, to perform computation to determine vacancies
- Identify parents who have not make appointments for open house
- Send email to each parent separately to inform on vacancy for open house event. The email recipient can later respond via email about their choice of open house dates.

These repetitive human tasks are now done by our email bot. The following flowchart illustrates the steps it will go through before sending an email with vacancies information.
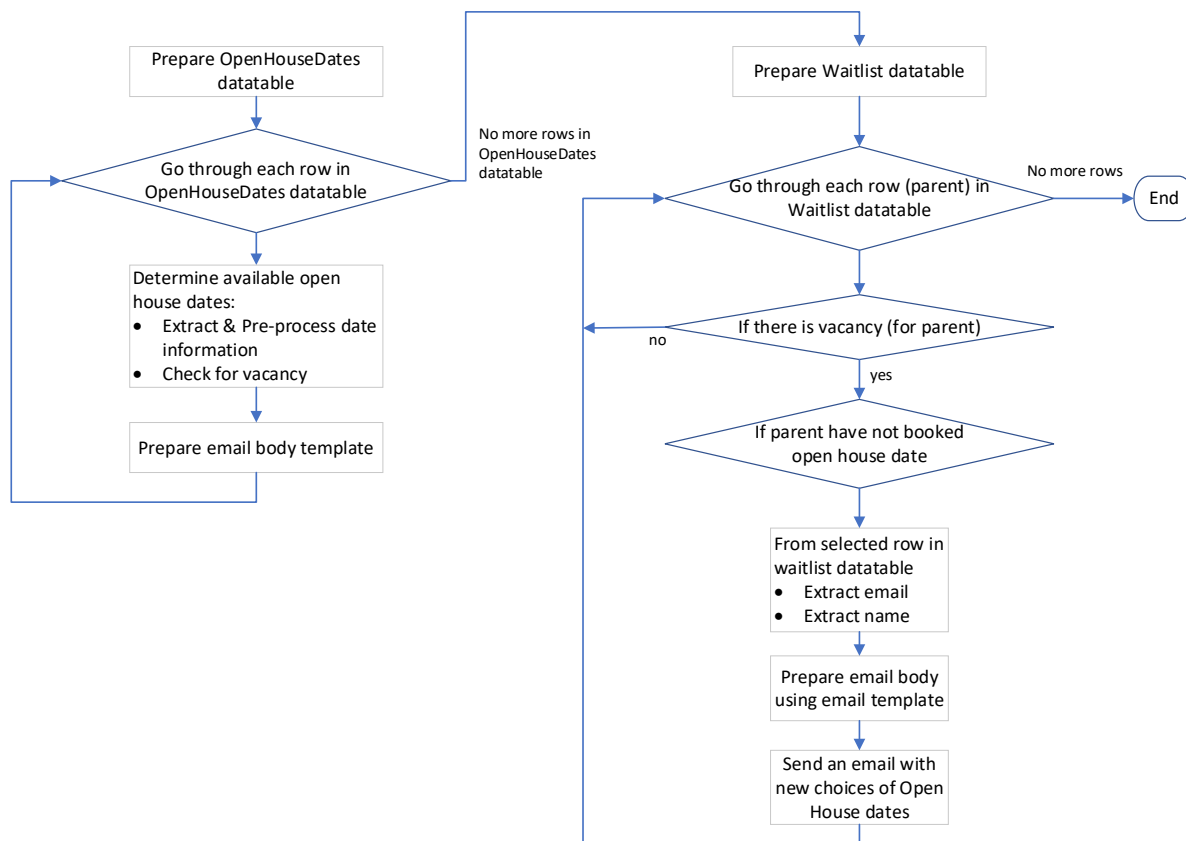

Figure 8: Inform parents on vacancies

There is a separate bot that processes email responses described under a section _5.1.5 Check Parents' Response, Update Waitlist, Obtain Insights_.

This email processing bot require parent's email responses to be standardized. Hence when the bot sends email to parents to inform on vacancies, there a small script behind each 'date of choice' hyperlink, which serves to generate standard response.

The parent can respond to this email in two mouse clicks: one click to choose an open house date, and another to click the 'send' button in their email tool.
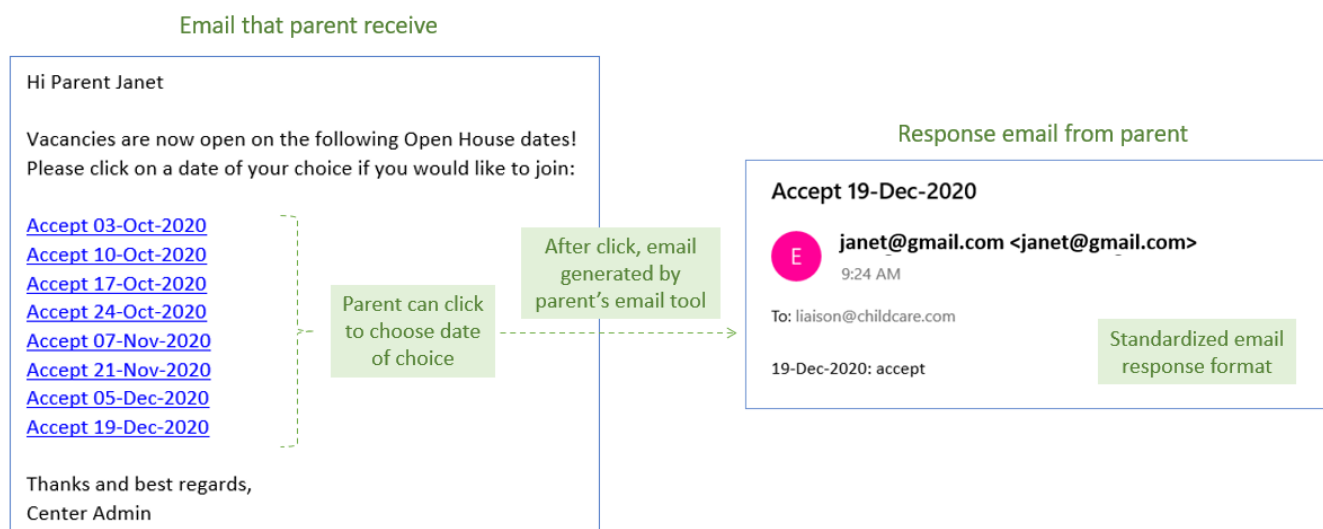

Figure 9: Parent's email receipt and response

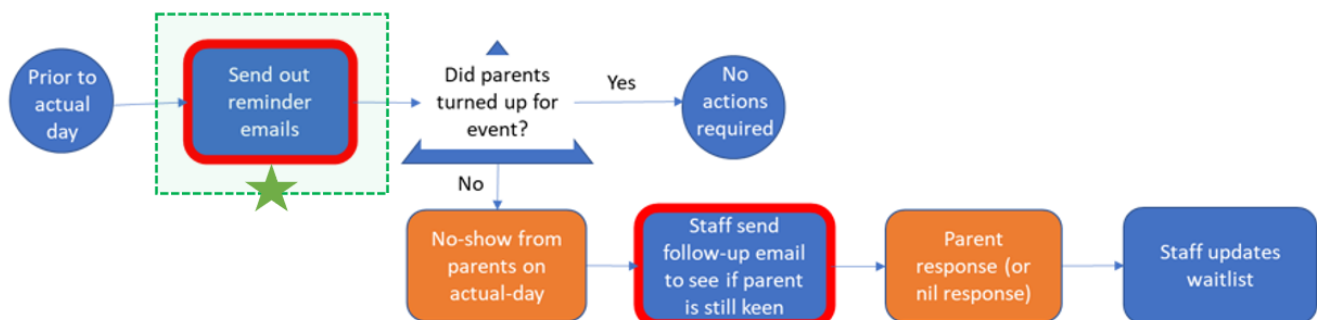## 5.1.3 Email Reminder on Open House Appointments

Figure 10: Workflow for Human Liaison

Open house appointments are often made by parents in advance (ie. a month or more) prior actual date of childcare open house event. There is a good chance that some parents will forget about the appointment.

A human liaison worker today has the repetitive tasks to:

- Go through a waitlist (from database) row by row, to determine what is the childcare open house event date that each parent had booked
- Send reminder email to each parent separately

Figure 11 illustrates the steps our bot will do to replace such repetitive human tasks:
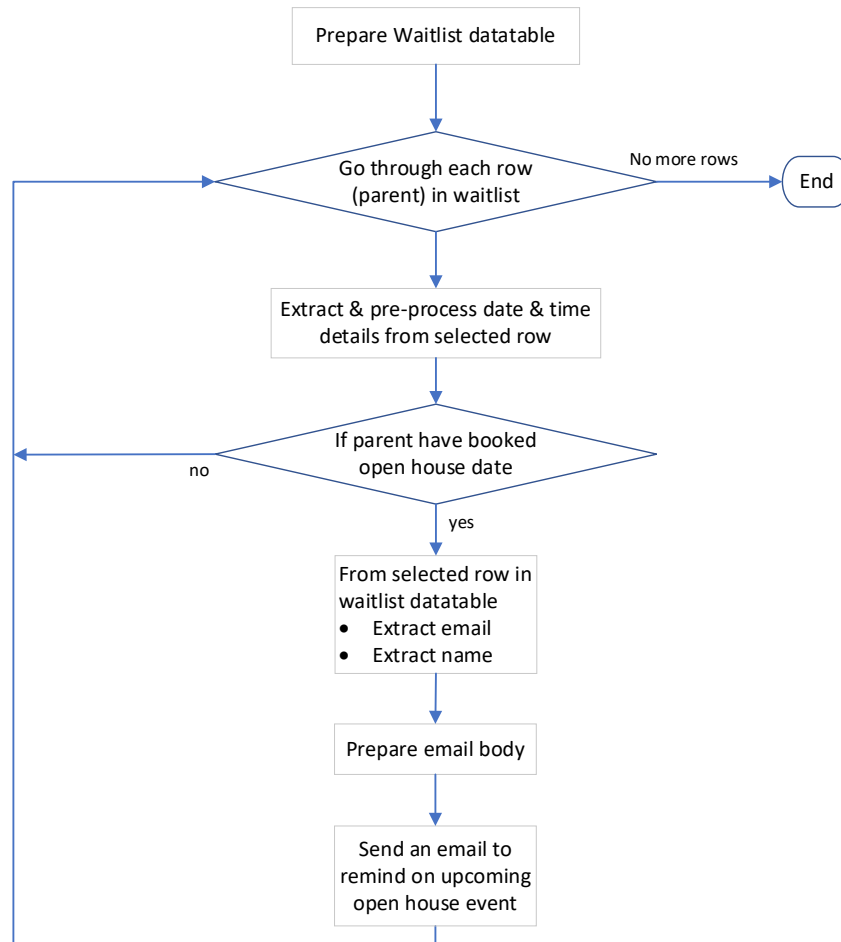

Figure 11: Reminder on open house appointments

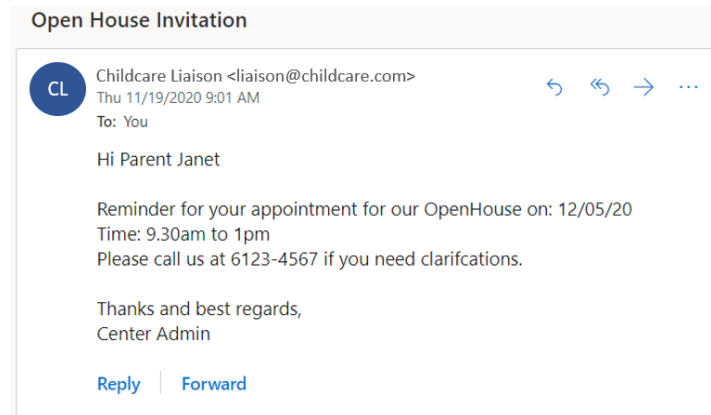Figure 12 illustrates how an email might look like for an email recipient.

Open House Invitation

CL   Childcare Liaison <liaison@childcare.com>
      Thu 11/19/2020 9:01 AM
      To: You

Hi Parent Janet

Reminder for your appointment for our OpenHouse on: 12/05/20
Time: 9.30am to 1pm
Please call us at 6123-4567 if you need clarifcations.

Thanks and best regards,
Center Admin

Reply | Forward

Figure 12: Parent's email reminder receipt

5.1.4 Email Follow-up with Parents on Missed Open House Event
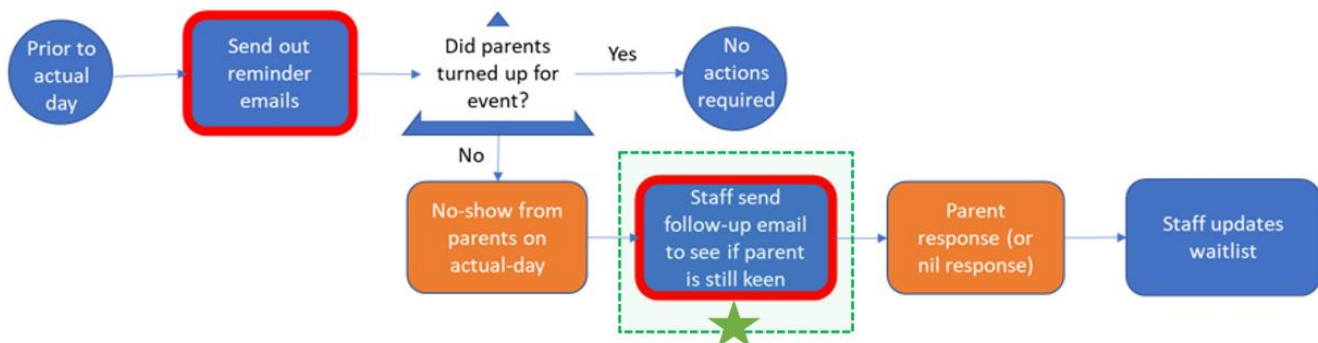

Figure 13: Workflow for Human Liaison

From time to time, there will be parents who do not turn up on the day of the open house. This is of high interest to childcare centers. There are many factors (e.g. distance from home, financial cost) a parent uses to look for a childcare. When a parent made the open house appointment with the childcare, it indicates that some factors had been fulfilled for them. These parents are hence prospects with high probability of conversion to customer. Childcare centers would want to try again to offer these prospects to come for another open house appointment.

A human liaison worker today will need to:

- Go through data tables (from database) containing information about vacancies row by row, to determine next available open house event dates.
- Go through a waitlist (from database) row by row, to identify parents who had missed their open house event appointment

- Send email to each parent separately to invite them to attend another open house event. The email will also seek to solicit feedback as to why parents do not choose to enroll their child in the childcare center should they decide not to attend another open house event.

Our bot replaces these repetitive human tasks. The following flowchart illustrates:
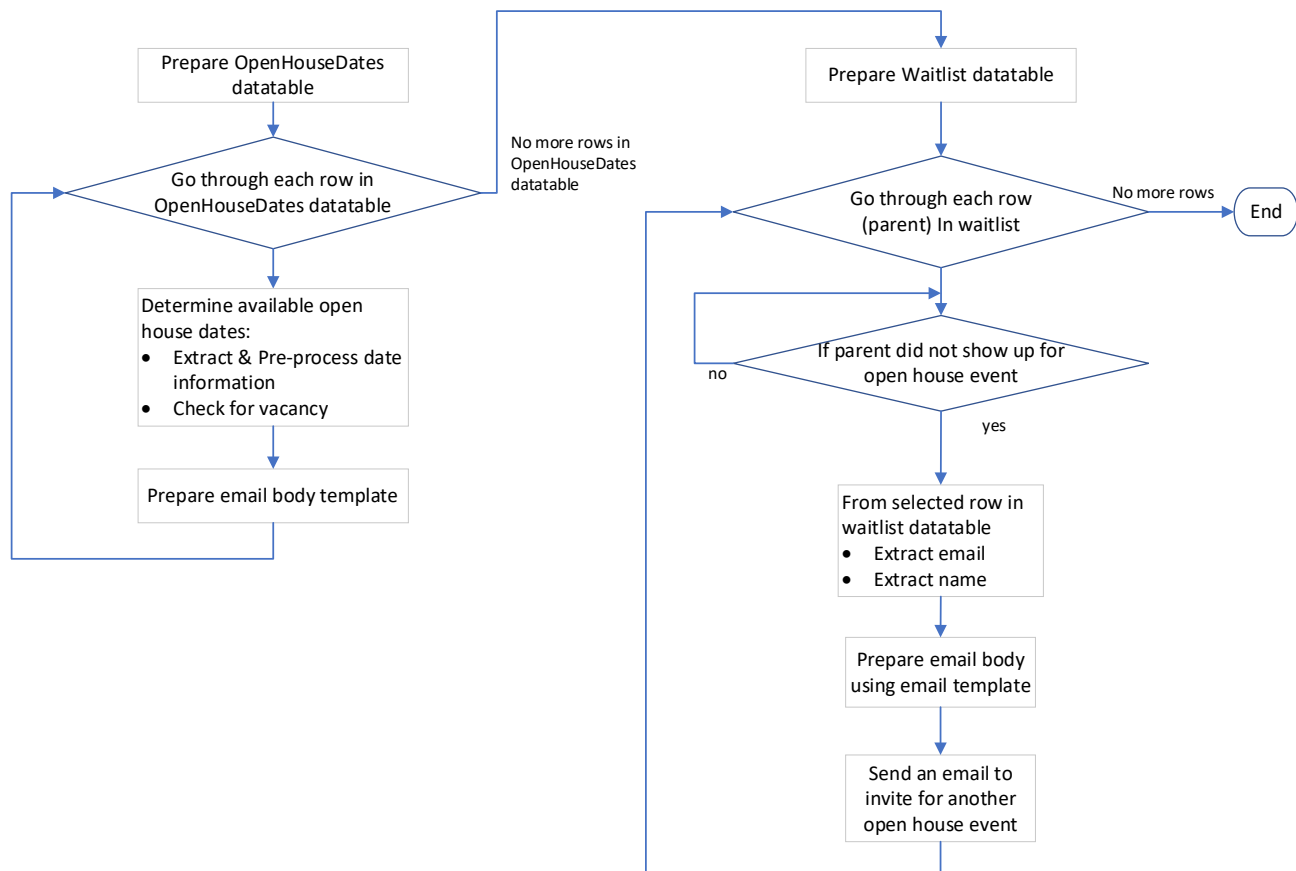

Figure 14: Follow up on parents who missed open house event

There is expectation for email responses from parents to this email re-invite. In an earlier section *5.1.2 Email to Inform Parents on Open* House Vacancies it was described that parent's email responses about their selection of open house dates needs be standardized. The same requirement applies here.

The parent can accept open house event invitations in two mouse clicks: one to choose a date, and another to click the 'send' button in their email tool.

Email that parent receive

Hi Parent Janet

We didn't have a chance to meet you at openhouse!

The following are our next available open house dates.

Would you like to choose a date that you wish to join? Please click on your choice and email us:

Accept 03-Oct-2020
Accept 10-Oct-2020
Accept 17-Oct-2020
Accept 24-Oct-2020
Accept 07-Nov-2020
Accept 21-Nov-2020
Accept 05-Dec-2020
Accept 19-Dec-2020

If you're not joining our next openhouse, for purpose of improving our services, may we seek your feedback on why we are not your choice for childcare centre?

Thank you!

Best regards,
Center Admin

Parent can click to choose date of choice

After click, email generated by parent's email tool

Response email from parent

Accept 19-Dec-2020

E janet@gmail.com <janet@gmail.com>
9:24 AM

To: liaison@childcare.com

19-Dec-2020: accept

Standardized email response format

<u>Figure 15: Parent's email receipt and response (for missed dates)</u>

5.1.5 Check Parents' Response, Update Waitlist, Obtain Insights

Prior to actual day → Send out reminder emails → Did parents turned up for event? — Yes → No actions required

No → No-show from parents on actual-day → Staff send follow-up email to see if parent is still keen → Parent response (or nil response) → Staff updates waitlist
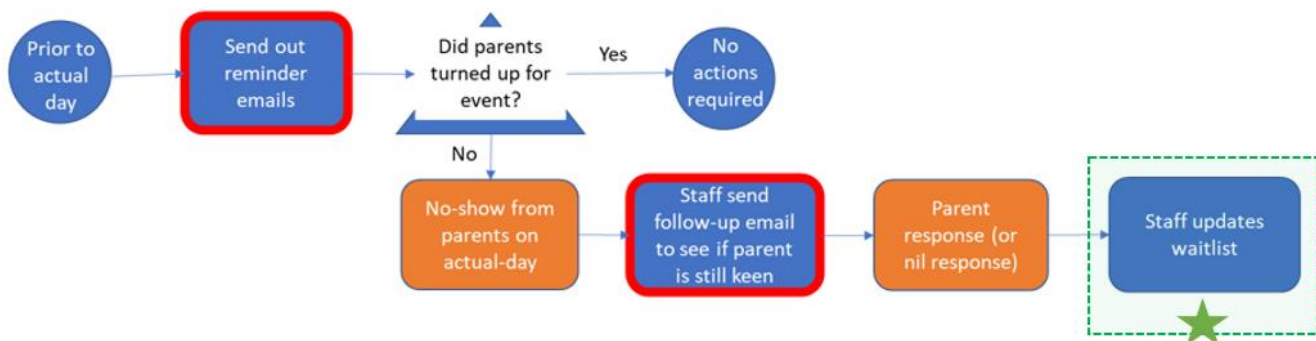
<u>Figure 16: Workflow for Human Liaison</u>

This process is active after emails had been sent out in an earlier process under section *5.1.4 Email Follow-up with Parents on Missed Open House* Event to re-invite parents for another open house event or to solicit feedbacks if the parent choose not to enroll their child in the childcare center.

A human liaison worker today needs to:

- Find emails that contain parents' responses in email inbox
- Categorize email responses into:
  1. Acceptance by parents to attend another open house event.
  2. Feedback about why a parent do not enroll their child in childcare center.

- If the email response is about (1), the human worker will update the waitlist about open house event date that parents selected
- If the email response is about (2), the human worker will try to understand the main points in parent's feedback, summarizes, and consolidate these points centrally (in an excel sheet eg.) for later use by childcare center to gain insights over consolidated feedback.

The following flowchart illustrates the steps our bot will go to replace these repetitive human tasks. Unread emails will be marked as read email in the email box by the bot:
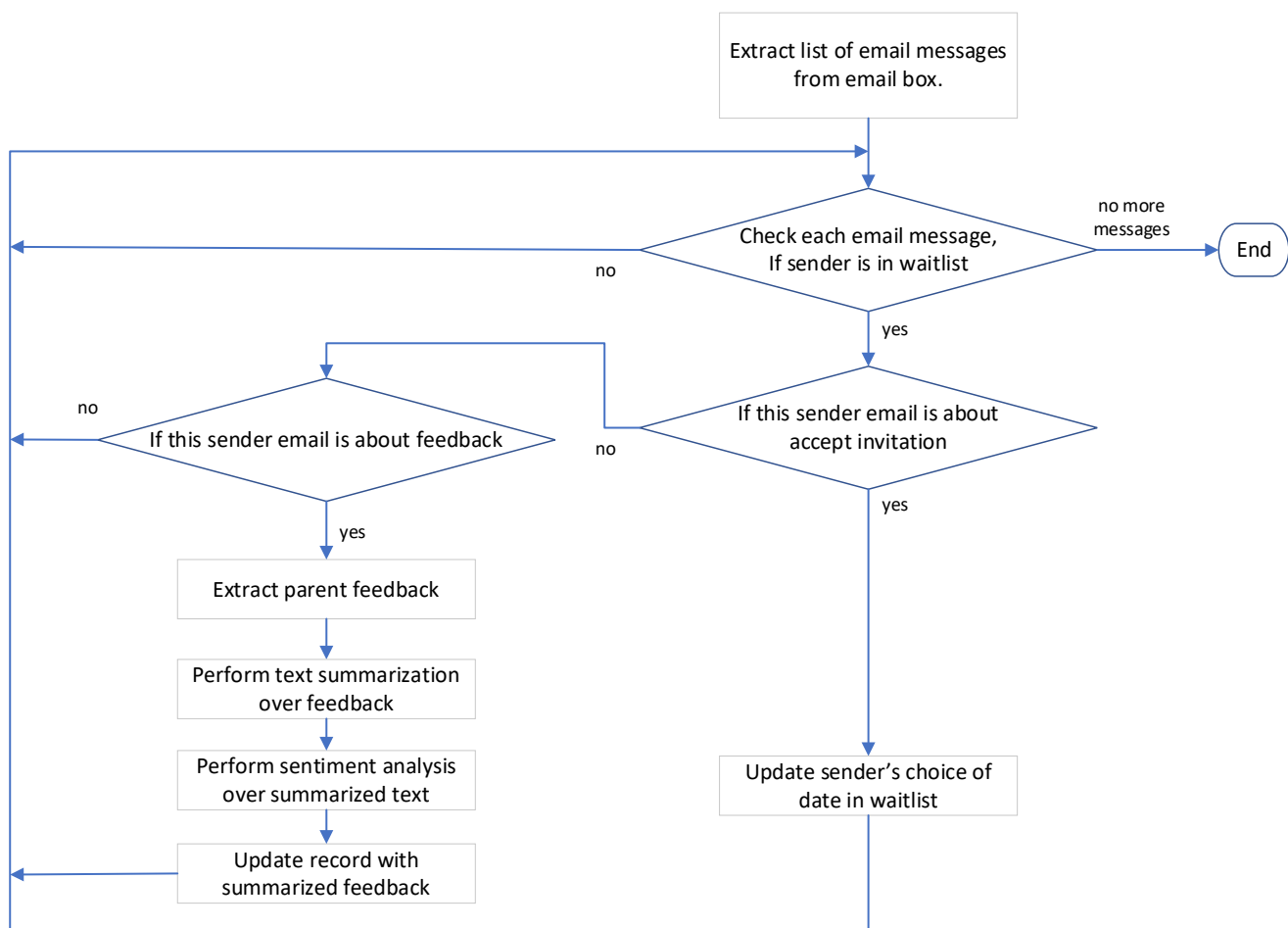

Figure 17: Check email response from parents

If parents responded with feedback, the bot will perform text summarization, apply sentiment analysis and consolidates these outputs in an Excel sheet. Hence, the bot transforms the unstructured natural language text from parents' feedback into a structured form in one spreadsheet.

Over time, as the structured data grows, the childcare center can perform pattern recognition tasks over this consolidated data to derive insights like top concern concerns among parents who chose not to enroll their child in the childcare center.

UIPath process is integrated with Python scripts containing functions running text summarization and sentiment analysis:

- The text summarization algorithm is based on the use of unsupervised learning to find text similarity. Ranking is then done to find most similar text and selected for text summary. The script is adapted from source: [text-summarizer](#)[4].
- VADER sentiment analysis library from NLTK was used to derive sentiment analysis of text. The sentiment score range from -1 to 1. A higher sentiment score represents a more positive sentiment.

The following is an example of what a consolidated spreadsheet of summarized feedback would look like.

email	date	feedback_summary	sentiment
janet@gmail.com	11/19/2020 13:19:17	,While i find the location and price to be ok, the operating hours of the childcare center does not meet my needs. The distance is actually further as compared to your location, but i need to go for that childcare centre because of their operating hours meet my needs. I was able to find another childcare center	0.1531
parentA@gmail.com	11/14/2020 17:51:48	, And to add on, I've also seen with my eyes the teachers in the childcare centre were very unprofessional in their behaviour towards the kids in the school. I also heard from my father-in-law that when he was peeking at my son thru the window of the school one day when he was eating his lunch that he actually picked up food which he dropped on the table to eat and there was not a single teacher who bothered to notice and stop him from doing so, until my father-in-law had to point it out to the teachers about it. And I also saw a Malay teacher flared her temper at a little gal who was crying by throwing her schoolbag on the floor	-0.9184
parentB@gmail.com	11/14/2020 13:53:40	So I would think every centre is different and wanting to switch to another Maplebear centre (hopefully with a better management). My girl has been in MapleBear TPY for a while now. She is happy going school there but issues started kicking in now they are approaching their first year anniversary	0.5106

Figure 18: Consolidated feedback

5.2 Chatbot

While an email bot was chosen to send out official info such as Open House invitations, a chatbot was created to handle ad-hoc requests a parent may have. It was, in general, more natural for an individual to turn to messaging apps for day-to-day ad-hoc communications.

Our chatbot supported the following ad-hoc requests:

---

[4] https://github.com/edubey/text-summarizer

1. Changing of Open House allocation
2. Getting your Open House allocation
3. Getting/ changing/ cancelling of available dates
4. Getting the event start time of your allocated Open House date
5. General enquires on center information

The chatbot frontend facing the user was the Telegram messaging app. This app was chosen as Telegram offered an official means to create chat bots, it also provided rich messaging features such as constructing MCQs, which we thought was ideal for soliciting structured responses from parents.

Messages incoming from the parents were piped into a chatbot backend written in python. These messages would be open-ended sentences written in natural language format, hence we leveraged on Cloud AI technologies (specifically Google's DialogFlow) to detect the intent and slots for a message. From there, should the intents required information from the central database or required the setting of certain values into the database, a DB Query module (written in python) would fetch/set the necessary information. Intent, slots, DB info were then returned to the Chatbot Backend for further processing and construction of a response which was shown to the user using Telegram.
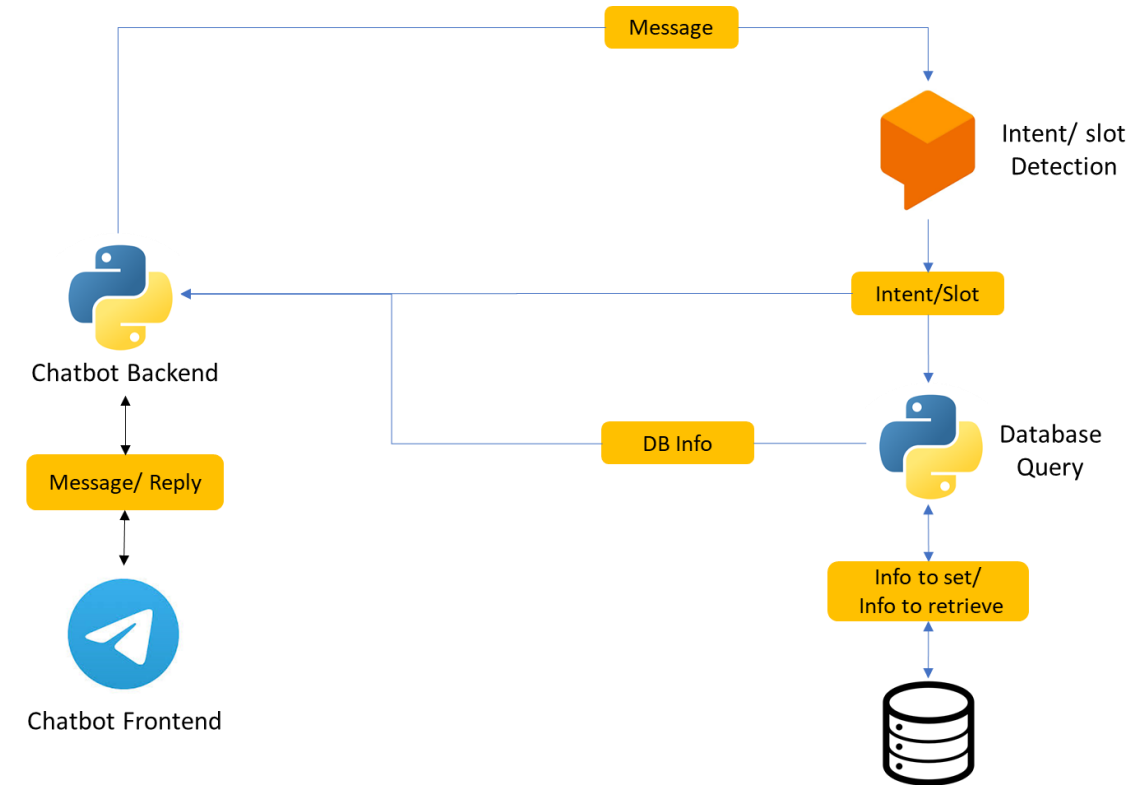
Figure 19: System Design of Chatbot

We show an example of a typical request (*change of availability*) from a parent to better illustrate our chatbot's workflow.
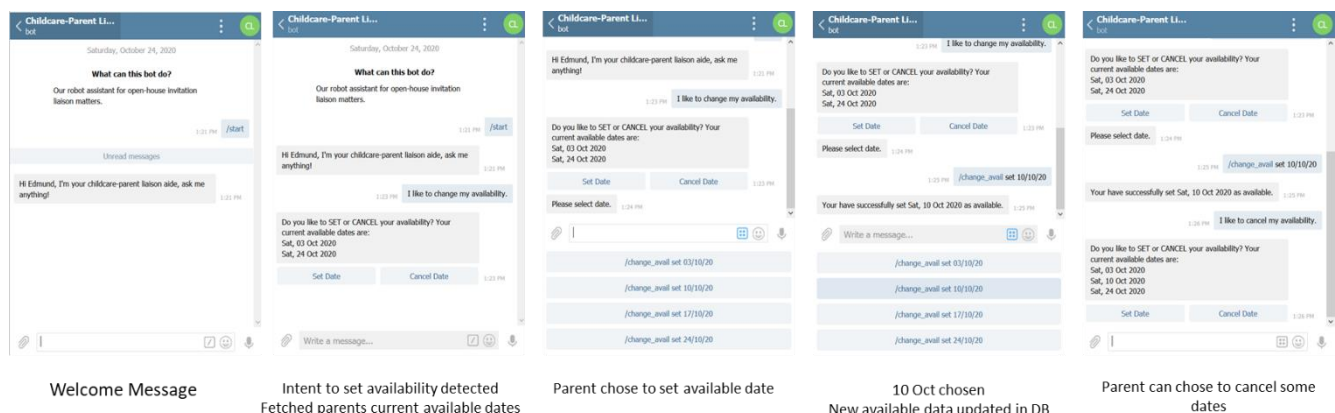

Figure 20: Chatbot Interaction Example (change of available date)

5.2.1 DialogFlow Submodule

The DialogFlow helps detect the intents and slots of the user request. The chatbot backend returns response with necessary information from analyzing the Intent, slots and DB info using Python. Some requests pull a direct reply from DialogFlow. The DialogFlow also helps answer user's general enquires about the center.

The intents and slots processed were as follows.

#	Intents	Default Response
1	welcome	Hi {firstname}, I'm your childcare-parent liaison aide, ask me anything!
2	register_user	Hi {firstname}, you have not set a telegram username or it is not the same one registered in our datebase. Fred not, you can still proceed by entering /register <youremail>, where <youremail> is the email used to correspond with our center.
3	register_success	Welcome {firstname}! You have been successfully registered to our chat aide.
4	register_fail	I'm sorry, {firstname}. Please check that the email was entered correctly and it was the one used in our prior correspondents, else please approach our center staff at <telephone number>.
5	alloc_get	Your allocated slot is {date}
6	alloc_get_fail_pending	We are still processing an allocation slot for you, kindly check back with us again tomorrow.
7	alloc_get_fail1	We were unable to allocate you an Open House slot, you may select a date below, or wait for the next month's dates.
8	alloc_get_fail2	We were unable to allocate you an Open House slot as there are no vacancies left. Kindly wait for the next month's dates.

9	alloc_get_fail3	You did not provide us your availability. Please select a date below, or wait for the next month's dates.
10	alloc_change_success	Your allocated slot has been successfully changed to [{new_date}]. The old slot on [{old_date}] has been cancelled.
11	alloc_change_fail_pending	You have not been allocated a slot yet. Changing of allocation slot is only allowed after you have been allocated one.
12	alloc_cancel_success	Your slot on {date} has been cancelled.
13	alloc_cancel_fail_syntax	We are unable to cancel your allocated slot, either you have no slots allocated or you should enter /cancel_alloc <allocated date> (in dd/mm/yy format).
14	alloc_cancel_fail_incorrect _date	Sorry, {date} is not your allocated slot.
15	avail_set	Do you like to SET or CANCEL your availability? Your current available dates are:
16	avail_set_success	You have successfully {action} {date} as available.
17	event_duration_user_alloc	Hi, your allocated Open House on {date} is open from {start} to {end}.
18	event_duration_user_ente red	Hi, our Open House date on {date} is open from {start} to {end}.
19	contConverse	Anything I can help you further with?
20	endConverse	Thank you from Childcare-Parent Liaison Aide! have a great day!
21	General_About	Our specialised curriculum is not based on subject matters, rather, all specialised areas work interdependently to ensure that every child is given numerous opportunities for challenge, risk-taking

		and social development by interacting with the natural world.
22	General_Facility	We provide Jungle Gym indoor playground as well as the aquatic environment for children to provide a fun and safe environment to enjoy the journey with us.
23	General_Operations	Monday – Friday: no later than 7 am / no earlier than 7.00 pm,Saturday : no later than 7 am / no earlier than 2.00 pm. Sundays and gazetted public holidays.half-day on the eves of any 3 public holidays another 6 days in a year, of which at least 2½ days are for staff training and development.
24	General_Process	The registration process will require the following: Proof of income/source of income, Health and Immunization form, Immunization Record for all children being enrolled, and Authorization for Emergency Medical Care. If your child requires medication, you will also need the Authorization for Dispensing Medication form.

Table 2: Intents and Response

6 System Performance & Validation

We validated that the RPA in our app was performing as designed with a Test Plan we had developed. Table 3 and Table 4 detailed the functional test cases we went through for the email bot and chatbot respectively.

ID	Test Case Description	Expected Result	Actual Result
1.	Verify that Open House invites are being sent by email bot 1. Keep xlsx db as-is 2. Click on Debug button in Main.xaml of PM3_UIPath_Send_Invites	Outlook draft folder should contain 2 new drafts ready to be send to Parent1, and Parent2	PASSED
2.	Verify that no Open House invites are sent if there are less than 3 dates with vacancies.	No new draft emails should appear in your outlook folder	PASSED

	1. In xlsx db, change column E so that less than 3 dates have vacancies 2. Click on Debug button in Main.xaml of PM3_UIPath_Send_Invites		
3.	Verify that parents' responses are updated in xlsx db 1. Using Parent2 (ipapm-ay2021@hotmail.com), reply to CC with "Y" to all dates 2. Click on Debug button in Main.xaml of PM3_UiPath_Receive_Invites	• parentAvailability sheet should be updated with Parent2's availability • waitingList sheet should have an OP date allocated to Parent2 • openHouseDates sheet should be updated (i.e. allocated date vacancy increment by 1)	PASSED
4.	Verify that "UnableToRetrieve" status is set if parent provides erroneous replies 1. Using Parent2 (ipapm-ay2021@hotmail.com), delete table of OP dates 2. Click on Debug button in Main.xaml of PM3_UiPath_Receive_Invites	waitingList sheet should have OP date of Parent2 updated with "UnableToRetrieve"	PASSED
5.	Verify that "NoDatesAvailable" status is set if there are no OP vacancies 1. Using Parent2 (ipapm-ay2021@hotmail.com), reply to CC with "Y" to all dates 2. In xlsx DB, openHouseDates sheet, change column E so that no vacancies are left 3. Click on Debug button in Main.xaml of PM3_UiPath_Receive_Invites	waitingList sheet should have OP date of Parent2 updated with "NoDatesAvailable"	PASSED
6.	Verify: if there is a parent in waitlist who has not booked for an open house event and there are vacancies, send email invitation to this parent. 1. In xlsx DB, waitingList sheet, prepare a row to include Parent2 (ipapm-ay2021@hotmail.com). Update the field 'openHouseDate' as either blank, "UnableToRetrieve" or "NoDatesAvailable". 2. Click on Debug button in Main.xaml of *VacancyAvailable_Inform Parent*	• Outlook draft folder should contain 1 new draft ready to be send to Parent2 • Draft email contains hyperlink to OP dates	PASSED
7.	Verify: if there is a parent in waitlist who had booked for an open house event, send reminder email to this parent. 1. In xlsx DB, waitingList sheet, prepare a row to include Parent2 (ipapm-ay2021@hotmail.com). Update the	Outlook draft folder should contain 1 new draft with reminder emails ready to be send to Parent2	PASSED

	field 'openHouseDate' with date 20/12/20. It can be other dates, but follow format dd/mm/yy. Ensure column in excel sheet is saved as text, not Microsoft Excel's Time format. 1. Click on Debug button in Main.xaml of *Reminder Email*		
8.	Verify: re-invitation emails are sent to parents with missed OP events 1. In xlsx DB, waitingList sheet, prepare a row to include Parent2 (ipapm-ay2021@hotmail.com). Update the field 'openhouseNoShow' with value YES. 2. Click on Debug button in Main.xaml of *NoShowParent Followup*	• Outlook draft folder should contain 1 new draft with emails ready to be send to Parent1 • Draft email contains hyperlink to next set of OP dates	PASSED
9.	Verify: After a parent sends email response about accepting re-invitation to open house event, email bot extracts information about OP date that parent selected, and update this in waitlist. 1. Use email account of Parent2 (ipapm-ay2021@hotmail.com). Reply to email that CC sent to Parent 2 under Test case 6. Reply by clicking on one of OP date choices. Send the email that was generated after clicking a date choice. 2. Check CC inbox (can use pm3-childcare@outlook.com) that Parent 2's email had been received. Ensure this email is marked as "Unread" 3. In xlsx DB, waitingList sheet, prepare a row to include Parent2 (ipapm-ay2021@hotmail.com). Update the field 'openHouseDate' as blank. 4. Click on Debug button in Main.xaml of *Email Processing_AcceptORFeedback*	In xlsx DB, waitingList sheet, the field 'openHouseDate' for Parent 2 is updated with parent's selected date choice.	PASSED
10.	Verify: text summarization and sentiment analysis is done for email responses from parents when the response is about feedback (on why they choose not to enroll child in CC) 1. Use email account of Parent2 (ipapm-ay2021@hotmail.com). Reply to email that CC sent to Parent 2 under Test case 6 with text: "Your fees are too high. I am unhappy with the fees. The location is ok for me. But I can find	• An existing spreadsheet named "feedback_consolidate.xls x" located in the same folder as *Processing_AcceptORFeed back UiPath workflow. This spreadsheet* is now updated with one additional new row. • In new row, under the field "feedback_summary",	PASSED

	another childcare center with lower fees." Note there are four sentences. 2. Check CC inbox (can use pm3-childcare@outlook.com) that Parent 2's email had been received. Ensure this email is marked as "Unread" 3. Click on Debug button in Main.xaml of *Email Processing_AcceptORFeedback*	there is update of two sentences of text. • Under the field "sentiment", there is update of a numeric value between range -1 to +1.	

Table 3: Test Plan for Email Bot

ID	Test Case Description	Expected Result	Actual Result
1.	Verify that parent can register himself with the chatbot 1. Trigger response with chatbot at t.me/myfirst_ay2021_bot using /start command 2. Reply chatbot with /register ipapm-ay2021@hotmail.com	Chatbot should reply: *"Welcome <your tg name>! You have been successfully registered to our chat aide."*	PASSED
2.	Verify that parent cannot register himself with the chatbot if email is incorrect 1. Revert xlsx DB so that tgUserId of Parent2 is a blank cell 2. Trigger response with chatbot using /start command 3. Reply chatbot with /register invalidemail@email.com	Chatbot should reply: *"I'm sorry, <your tg name>. Please check that the email was entered correctly and it was the one used in our prior correspondents, else please approach our center staff at <telephone number>.*	PASSED
3.	Verify that parent can get allocated OP date from chatbot 1. In xlsx DB, waitingList sheet, col I, enter "03/10/20" in string format for Parent2 2. Reply chatbot with "Get Allocation"	Chatbot should reply: *"Your allocated slot is Sat, 03 Oct 2020"*	PASSED
4.	Verify that parent can change allocated OP date 1. Reply chatbot with "Change Allocation" 2. Options with "Change Date" and "Cancel Date" should appear 3. Tap on "Change Date" 4. List of OP dates should appear 5. Tap on option with "10/10/20"	• Chatbot should reply: *"Yout allocated slot has been successfully changed to [Sat, 10 Oct 2020]. The old slot on [Sat, 03 Oct 2020] has been cancelled"* • Open xlsx DB, and openHouseDate sheet, 03/10/20 should should 0 vacancyAllocated, 10/10/20 should show 1	PASSED
5.	Verify that parent can set available dates 1. Reply chatbot with "Set Availability"	• Chatbot should reply: "You've successfully set <date you tapped on> as available"	PASSED

	2. Options to "Set Date" or "Cancel Date" should appear 3. Tap on "Set Date" 4. List of OP dates that you can set as available should appear 5. Tap on all dates	• Open xlsx DB, and parentAvailability sheet, parentId=2 should have a row of ones	
6.	Verify that parent can set available dates 1. Reply chatbot with "Set Availability" 2. Options to "Set Date" or "Cancel Date" should appear 3. Tap on "Cancel Date" 4. List of OP dates that you can cancel should appear 5. Tap on all dates	• Chatbot should reply: "You've successfully cancel <date you tapped on> as available" • Open xlsx DB, and parentAvailability sheet, parentId=2 should have a row of zeroes	PASSED
7.	Verify that parent can get start-end time of event 1. Reply chatbot with "Get open house time"	• Chatbot should reply: "Hi your allocated Open House on <data> is open from <start time> to <end time>	PASSED
8.	Verify that parent can change allocated OP date 1. Reply chatbot with "Change Allocation" 2. Options with "Change Date" and "Cancel Date" should appear 3. Tap on "Cancel Date" 4. Reply chatbot with "/cancel_alloc <date in dd/mm/yy format>" e.g. 03/10/20	• Chatbot should reply: *"Your slot on <date> has been cancelled"*	PASSED
9.	Verify that parent can select an OP date if you have cancelled one before 1. Ensure you have no allocated OP date (go through s/n 15 to ensure this) 2. Reply chatbot with "Select allocation" 3. List of OP dates that you can choose from should appear 4. Tap on 03/10/20	• Chatbot should reply: *"Yout allocated slot has been successfully changed to [Sat, 3 Oct 2020]. The old slot on [None] has been cancelled"*	PASSED

Table 4: Test Plan for Chatbot

7 Conclusion

In this project we developed RPA to support the liaison efforts between a Childcare Center admin and prospective parents. We leveraged upon the rich features from UiPath to quickly automate 5 sets of workflow involving MS Excel and MS Outlook. This helped saved hours daily and eliminated the human errors that often arise due to repetitive mundane work.

To enhance our user's experience (both childcare center admin and parents), we leveraged on CloudAI technology (Google Dialogflow) to build a chatbot that can answer to 5 different types of ad-hoc request. This helped eased the center admin from being bothered by the constant stream of ad-hoc requests, and gave parents the gratification of a 24/7 helpdesk service and instant replies to their queries.

7.1 Limitations and Improvements

- When parents reply to email bot on the Open House dates which they are available, it is expected that they reply in a table in the email with "Y" next to the dates they are available and "N" for dates they are not. This restricts the effectiveness of the email bot. It would be good if voting buttons can be used in the next iteration.

1 Installation and User Guide

1.1 Setup for Email Bot

1. This installation instruction is written on a Windows 10 OS. Outlook app should already be pre-installed.
2. Clone the source code from the following GitHub link: https://github.com/chen-mingyie/childcare-parent-liaison-aide/
3. Add an email account to your Outlook app following instructions in this page under "Add your Outlook.com email account quickly". You can use our mock childcare Outlook account, pm3-childcare@outlook.com. The password is "pm3childcare2020". If for whatever reason you cannot use this mock account, you can use any other Outlook account as well.
4. An email account acting as a pseudo-parent was set up for this project, email address = ipapm-ay2021@hotmail.com, email password = IPAagent789;
5. Leave your Outlook app open. Next, download UiPath.
6. Retrieve this project's UiPath files from *SourceCode/UiPath*. There are a total of 6 different UiPath projects in .zip format. Unzip the project files, ensure each project follows the folder structure below: for example

        SourceCode
        |------ UiPath
        |-----------PM3_UiPath_Recieve_Invites
        |---------------- project.json is found here

Name	Date modified
Email Processing_AcceptORFeedback	23/11/2020 11:09 am
NoShowParent Followup	23/11/2020 11:09 am
PM3_UiPath_Receive_Invites	23/11/2020 11:09 am
PM3_UIPath_Send_Invites	23/11/2020 11:09 am
Reminder Email	23/11/2020 11:09 am
VacancyAvailable_Inform Parent	23/11/2020 11:09 am
UiPath_Projects.zip	23/11/2020 11:07 am
UiPath Project Unzipping Instructions.txt	23/11/2020 11:08 am

cmingyi > PycharmProjects > childcare-parent-liaison-aide > SourceCode > UiPath > PM3_UIPath_Send_Invites

Name	Date modified	Type	Size
.entities	16/10/2020 11:37 am	File folder	
project.json	18/11/2020 6:17 pm	JSON File	2 KB
email_html.txt	22/11/2020 9:49 am	Text Document	3 KB
Get database from Excel.xaml	22/11/2020 10:33 am	Windows.XamlDo...	17 KB
Main.xaml	23/11/2020 9:34 am	Windows.XamlDo...	13 KB

7. Open the UiPath file using UiPath studio. First, double click "Open a Local Project". Next, go to the location you left the UiPath files and select "project.json" and click "Open".
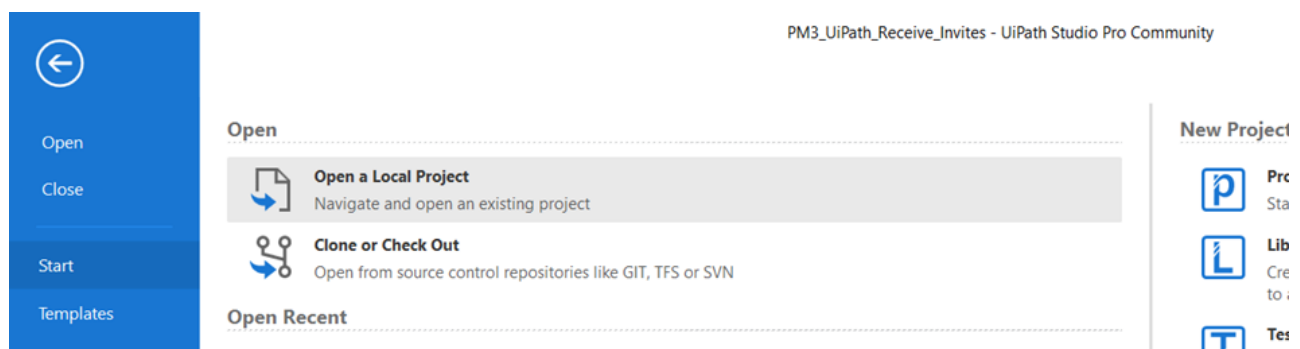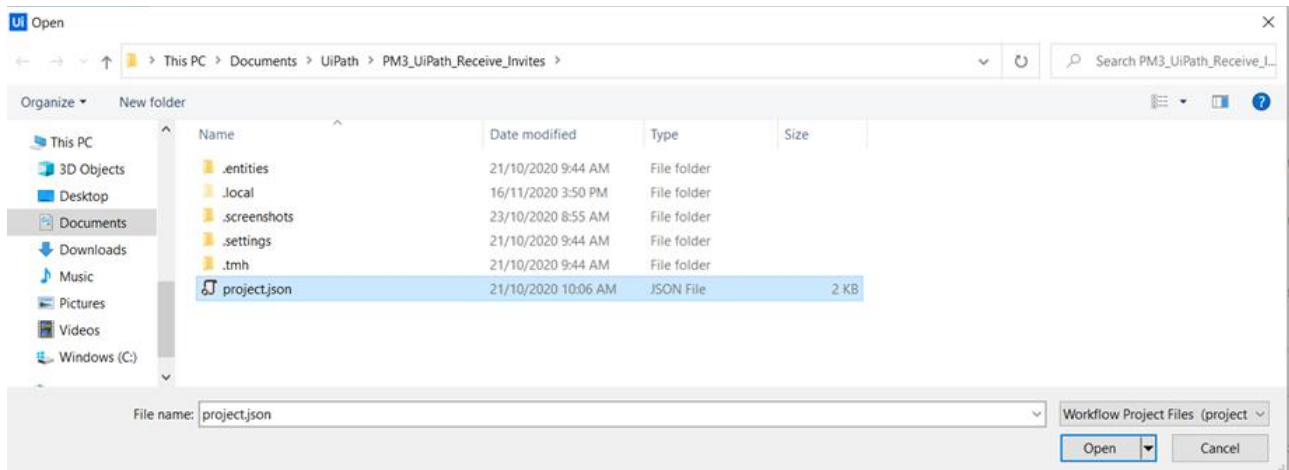
8. If you use any account other than pm3-childcare@outlook.com, please change the variable "OutlookAccount" circled in red to your Outlook account in quotes.
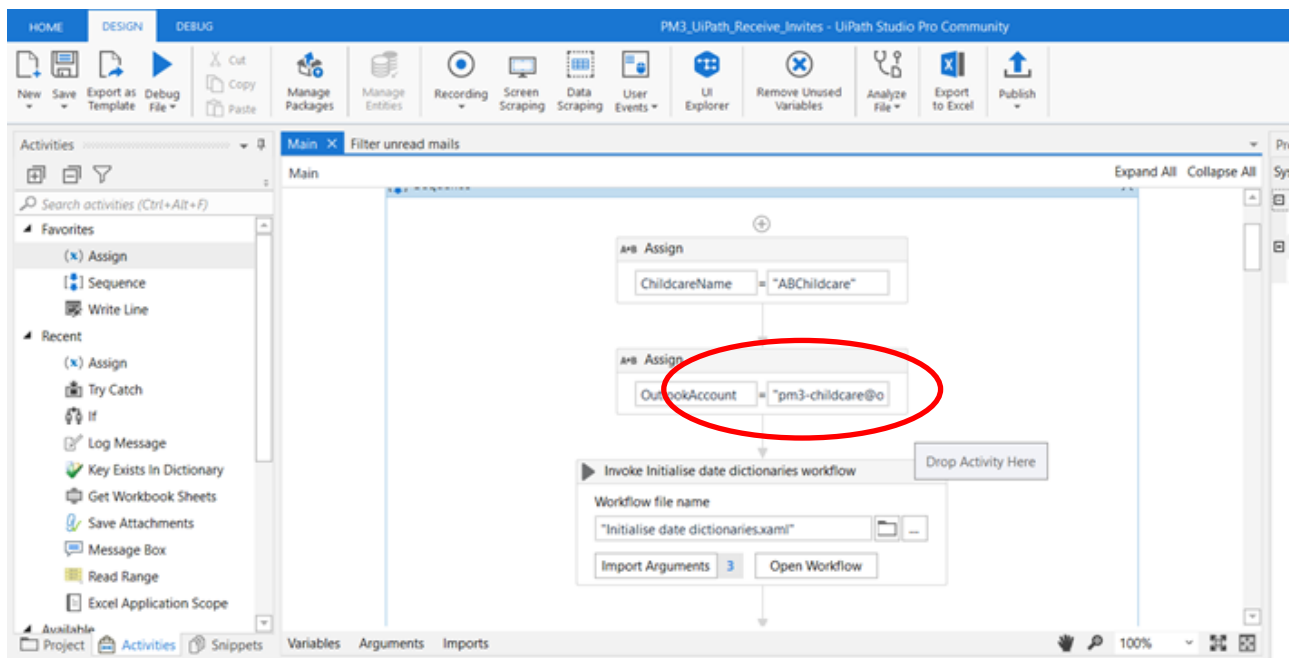
1.1.1 Preparing UiPath to work with Python environment

Preparing Python environment

It was observed via trial and error in this project that UiPath Studio has following limitations:

i. UiPath Studio needs to work with python.exe in a <u>non-virtual</u> python environment.

ii. The latest version that UiPath Studio can work with is Python 3.6.X.

Two steps to prepare Python environment

1. Install python in your machine:
    a) Python version 3.6., <u>Windows x86-64 executable installer</u>: https://www.python.org/ftp/python/3.6.6/python-3.6.6-amd64.exe
    b) If you wish to use other Python versions, the following contains installers for other versions: https://www.python.org/downloads/

2. Install three Python libraries: *numpy, nltk, networkx*:
    a) At command prompt, access folder when you had just installed Python.
    b) Keep a record this file path the python executable. There is need to use this file path later at UiPath Studio.

c) Access the folder "Scripts"
d) Install the following Python libraries using the following: **pip install numpy nltk networkx**

```
Command Prompt                                              —    □    ×

C:\Users\edang\AppData\Local\Programs\Python\Python36>cd Scripts

C:\Users\edang\AppData\Local\Programs\Python\Python36\Scripts>pip install numpy nltk networkx
```

Preparing UiPath Studio environment to use Python

There are two UiPath Studio files that needs to be updated to specify which Python environment to use:

   i.   'text_summarize.xaml'
    ii.   'text_sentiment.xaml'

There are two Python scripts that needs to be loaded into the two respective UiPath Studio file mentioned above

    i.   'text_summarize.py'
    ii.   'text_sentiment.py'

All these files are located under the folder *<Feedback_Why_No>*.

Two steps to prepare UiPath Studio environment to use Python

    1. Update file paths in 'text_summarize.xaml'
        a. Open file 'text_summarize.xaml' in UiPath Studio
        b. Specify python environment for 'text_summarize.xaml'. See green comments in diagram
        c. Specify folder location of 'text_summarizer.py'. See red comments in diagram
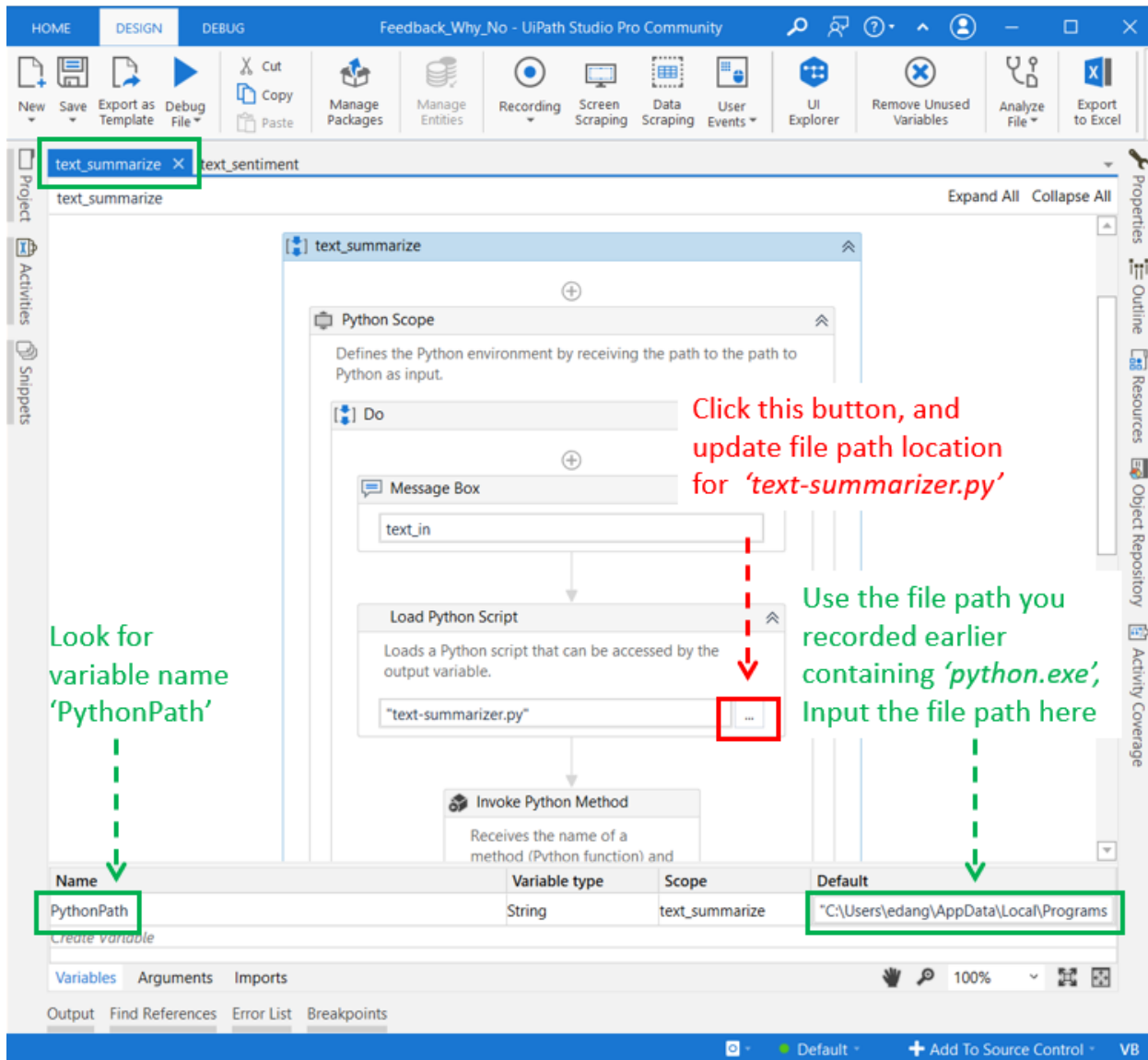
2. Update filepaths in 'text_sentiment.xaml'
   a) Open file 'text_sentiment.xaml' in UIPath Studio
   b) Specify python environment for 'text sentiment.xaml'. See green comments in diagram
   c) Specify folder location of 'text sentiment.py'. See red comments in diagram

1.2 Setup for Chatbot

1. This installation instruction is written on a Windows 10 OS.
2. Clone the source code from the following GitHub link: https://github.com/chen-mingyie/childcare-parent-liaison-aide/
3. Download and install Anaconda Navigator via the link https://www.anaconda.com/products/individual
4. Download and install Pycharm IDE via the link https://www.jetbrains.com/pycharm/
5. Install Telegram on your iOS or Android mobile phone. Sign up for an account.
6. Open Pycharm and go to File → Open → Select the folder ISA-IPA-2020-11-23-IS02FT-GRP-Childcare-Parent-Liaison-Aide. The contents of the folder should have the following

Name
Miscellaneous
ProjectReport
SourceCode
Video
.gitignore
README.md
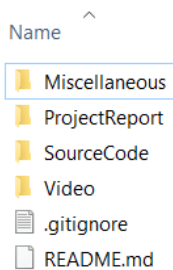
7. Go to File → Settings. On the left-hand panel select "Project: <your project name>", click on "Python Interpreter", click on the settings icon ⚙
8. Click on Add. Click on Conda Environment. Click on Python version and select 3.7. Click on OK.
9. The demo requires the following packages, do a *pip install <package name>* of them to the base venv if they are not already installed:
   - *flask==1.1.2*
   - *dialogflow==1.1.0*
   - *python-telegram-bot==13.0*
   - *pandas==1.1.3*
   - *xlrd >= 1.0.0*
   - *xlsxWriter==1.3.7*

10. [Ignore this point if you have downloaded the source code from Luminus] The json key to access Google CloudAI for DialogFlow is not published in GitHub (it should not). Retrieve the json key (*google-credentials.json*) which is uploaded to Luminus and place it in the folder: *SourceCode\app\business*

11. [Ignore this point if you have downloaded the source code from Luminus] The secret token to give full access to the Telegram bot is not published in GitHub (it should not). Retrieve the secret token (*secret_idandkeys.py*) which is uploaded to Luminus and place it in the folder: *SourceCode\app\model*

12. Start the chatbot backend by going to SourceCode → start_chatbot_backend.py. Click on Run → Run → start_chatbot_backend.py.
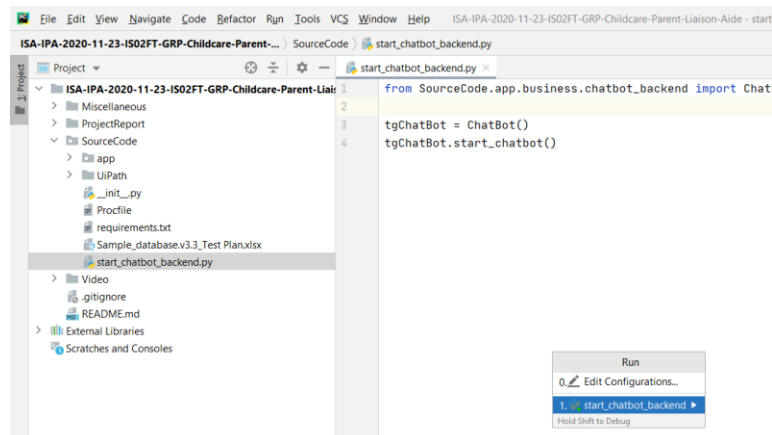
13. Seeing the log: apscheduler.scheduler – INFO – Scheduler started signals that the chatbot backend had been successfully started

14. The chatbot can be found by searching for *@myfirst_ay2021_bot* in the Telegram search bar, or by navigating to t.me/myfirst_ay2021_bot in a web browser.
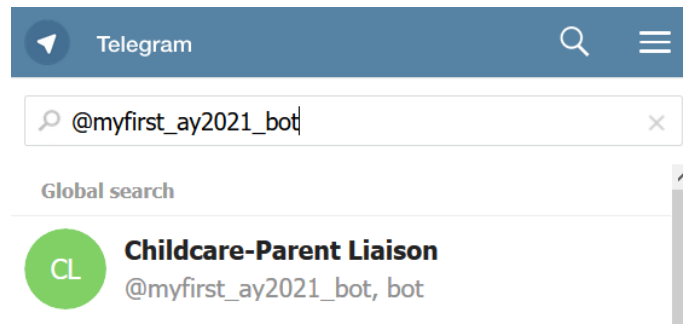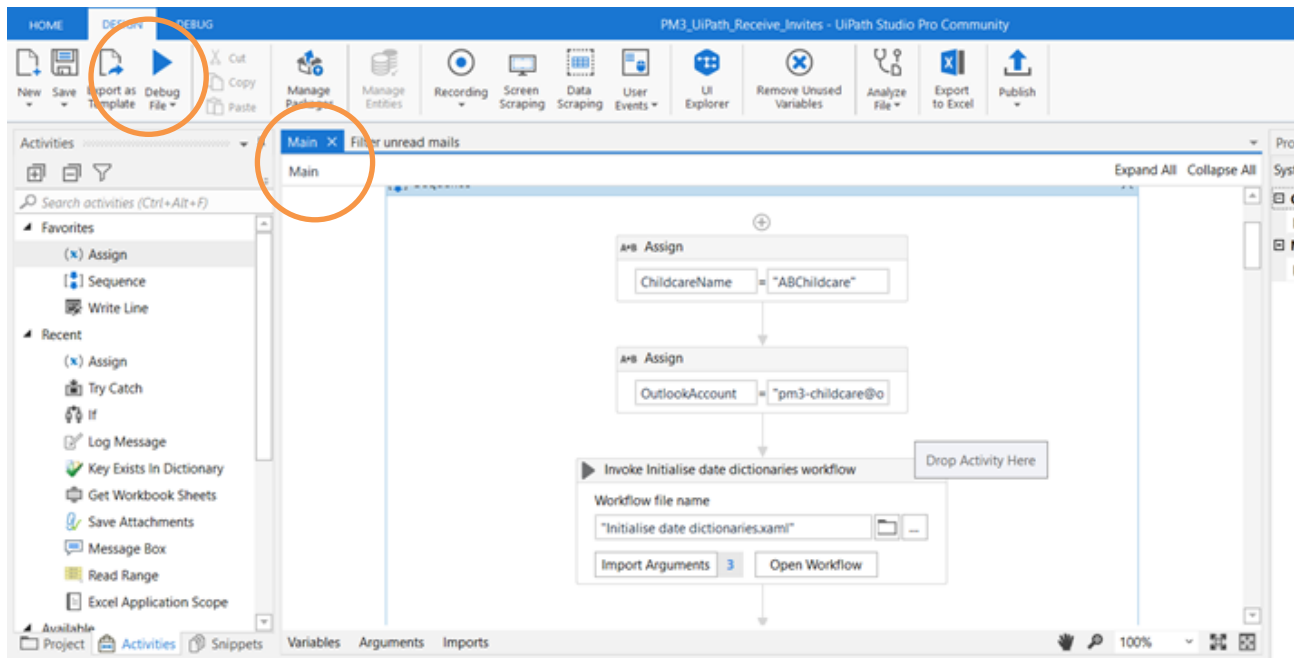

15. To terminate the chatbot backend, click on the stop button, you should see this in the console.

1.3 User Guide on Email Bot

With your Outlook app open and the Outlook account signed in (refer to this section 1.1 Setup for Email Bot step 6 on the account to be signed in), click "Debug File" while at Main.xaml (Main.xaml is highlighted).
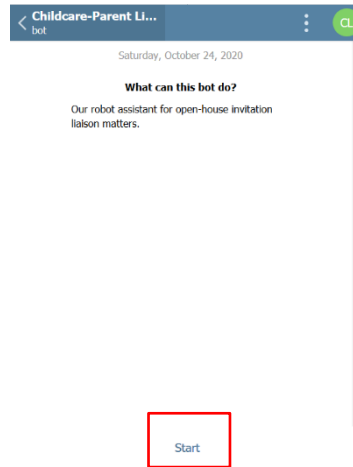

Email bot 'Send emails' and 'Receive emails' sections

Please note that emails will be kept as Drafts instead of sending out automatically, this is to enable childcare operators to check the drafts before sending out and give them an ease of mind.

1.4 User Guide on Chatbot

1. Click on *start* button to begin your interaction with the chatbot.

2. First time user of the chatbot will need to register themselves. To do so enter the command: */register <email>* into Telegram, choose ipapm-ay2021@hotmail.com when testing our app. The application's database is found under the filepath: *SourceCode\Sample_database.v3.3_Test Plan.xlsx*

3. The chatbot supports the following tasks

   3.1   Changing of Open House allocation (e.g. "Change allocation")

   3.2   Getting your Open House allocation (e.g. "Get allocation")

   3.3   Getting/ changing/ cancelling of available dates (e.g. "Change availability")

   3.4   Getting the event start time of your allocated Open House date (e.g. "Get event duration")

   3.5   General enquires on center information

   You may use the example sentences or form your own sentences to enquire on any of the above tasks, then follow the on-screen instructions for each task.