NTU Mod Swap bot

@NTUModSwap_bot

BY TEAM SCRUBZBOT

CZ1003 ASSIGNMENT

Members:

- Lee Wei Min (U1720166K)
- Bryan Leow Xuan Zhen (U1721837L)
- Dennis Christopher Suherman (U1722583E)
- Kong Jie Wei (U1720017C)
- Priscilla Teo Qiu Yee (U1720076L)

Contents

Problem Statement and Objective 3
Current Tools Available and Analysis 3
1. HardWareZone (HWZ) “Official NTU Swap Course Index Thread” 3
2. NBS ABC ADD/DROP Excel Spreadsheet 4
Significance of our problem 5
Our proposed solution 5
1. Check various indexes for the course 5
2. Input entry to swap course index: 6
3. Check pending status in index swapping 7
4. Cancel pending status in index swapping 7
Service Integration 8
1. Web Scraping 8
2. Email API (Sending) 8
3. Email API (Validation) 8
How is our idea better than current tools available? 9
Problems we faced during the project: 9
1. Unit counter problem 9
2. Waiting time for server email validation 10
3. Logic problem for courses with one index only 11
Conclusion/Moving on 12
Appendix 13
Survey results 13
Screenshots of our Telegram bot: 14
Flowchart 18
User Acceptance Testing 20

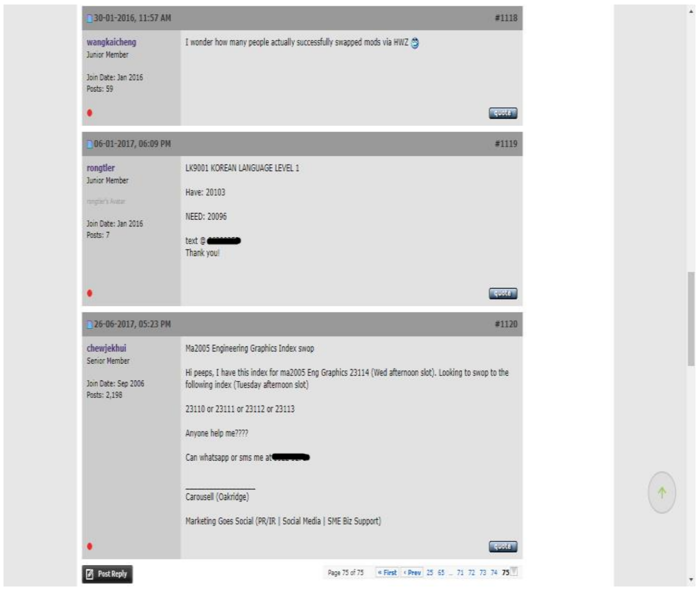
Problem Statement and Objective

During the add-drop period at the start of every semester in NTU, undergraduates find it hard to secure their preferred timeslot for the courses that are they taking in that semester. Students may choose to wait for vacancies for their preferred index or in very rare and compelling cases (such as final year students who need to clear all their modules to graduate on time), write to the school administration to seek assistance in securing their preferred index. For most students, it is more practical to find another student to swap their course index with for both students to secure their preferred timings without having to fight for vacancies available for their preferred course index. Unless undergraduates can identify someone whom they can swap their course index with, they are otherwise left with two choices--either to compromise on their non-academic commitments or to drop those courses that clashes with their commitments.

We, TeamScrubzBot are formed with a sole purpose in mind - to eradicate the problem faced by NTU undergraduates once and for all by offering an interactive and effective platform to facilitate the swapping of class index.

Current Tools Available and Analysis

1. HardWareZone (HWZ) "Official NTU Swap Course Index Thread"



30-01-2016, 11:57 AM #1118

wangkaicheng
Junior Member
Join Date: Jan 2016
Posts: 59

I wonder how many people actually successfully swapped mods via HWZ

06-01-2017, 06:09 PM #1119

rongtler
Junior Member
rongtler's Avatar
Join Date: Jan 2016
Posts: 7

LK9001 KOREAN LANGUAGE LEVEL 1

Have: 20103

NEED: 20096

text @[redacted]
Thank you!

26-06-2017, 05:23 PM #1120

chewjekhui
Senior Member
Join Date: Sep 2006
Posts: 2,198

Ma2005 Engineering Graphics Index swop

Hi peeps, I have this index for ma2005 Eng Graphics 23114 (Wed afternoon slot). Looking to swop to the following index (Tuesday afternoon slot)

23110 or 23111 or 23112 or 23113

Anyone help me????

Can whatsapp or sms me at[redacted]

Carousell (Oakridge)

Marketing Goes Social (PR/IR | Social Media | SME Biz Support)

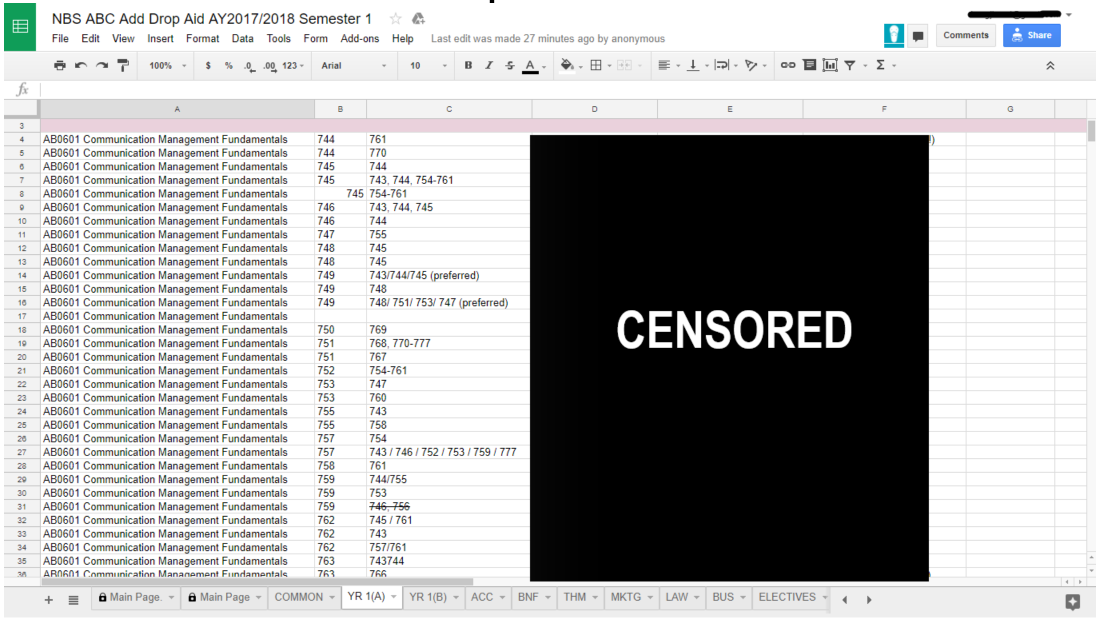
Figure 1: Example of a HardWareZone index swapping thread

In HWZ, there is a forum relating to university academic matters. One of the threads allows any users to publicly post their willingness to swap their course index. If there are 2 posts that have complementary data[1], the users may contact each other and proceed to swap

[1] We define complementary data as a pair of data by 2 users, where user A has index xxxxx and wants index yyyyy, while user B has index yyyyy and wants index xxxxx.

their index in the official NTU's STARS website. Currently, this is the only tool that is available to NTU students from all faculties to conduct their index swapping.
However, the HWZ forum is not able to directly link students up for index swapping, even if there is a pair of entries that complement with each other. The students will have to regularly check the forum for any new entries that has the index that they want. Oftentimes, students do not check the forum, and is likely to expect the other party to contact them as they left their contact available on the forum. There is also an issue with finding the posts that is relevant to the students amongst the many other posts that are posted by students of a different faculty. Even if the search function is used, as users may freely write their posts, it may not be consistent amongst users[2].

2. NBS ABC ADD/DROP Excel Spreadsheet


NBS ABC Add Drop Aid AY2017/2018 Semester 1

Last edit was made 27 minutes ago by anonymous

[redacted]

	A	B	C
3			
4	AB0601 Communication Management Fundamentals	744	761
5	AB0601 Communication Management Fundamentals	744	770
6	AB0601 Communication Management Fundamentals	745	744
7	AB0601 Communication Management Fundamentals	745	743, 744, 754-761
8	AB0601 Communication Management Fundamentals	745	754-761
9	AB0601 Communication Management Fundamentals	746	743, 744, 745
10	AB0601 Communication Management Fundamentals	746	744
11	AB0601 Communication Management Fundamentals	747	755
12	AB0601 Communication Management Fundamentals	748	745
13	AB0601 Communication Management Fundamentals	748	745
14	AB0601 Communication Management Fundamentals	749	743/744/745 (preferred)
15	AB0601 Communication Management Fundamentals	749	748
16	AB0601 Communication Management Fundamentals	749	748/ 751/ 753/ 747 (preferred)
17	AB0601 Communication Management Fundamentals		
18	AB0601 Communication Management Fundamentals	750	769
19	AB0601 Communication Management Fundamentals	751	768, 770-777
20	AB0601 Communication Management Fundamentals	751	767
21	AB0601 Communication Management Fundamentals	752	754-761
22	AB0601 Communication Management Fundamentals	753	747
23	AB0601 Communication Management Fundamentals	753	760
24	AB0601 Communication Management Fundamentals	755	743
25	AB0601 Communication Management Fundamentals	755	758
26	AB0601 Communication Management Fundamentals	757	754
27	AB0601 Communication Management Fundamentals	757	743 / 746 / 752 / 753 / 759 / 777
28	AB0601 Communication Management Fundamentals	758	761
29	AB0601 Communication Management Fundamentals	759	744/755
30	AB0601 Communication Management Fundamentals	759	753
31	AB0601 Communication Management Fundamentals	759	~~746, 756~~
32	AB0601 Communication Management Fundamentals	762	745 / 761
33	AB0601 Communication Management Fundamentals	762	743
34	AB0601 Communication Management Fundamentals	762	757/761
35	AB0601 Communication Management Fundamentals	763	743744
36	AB0601 Communication Management Fundamentals	763	766

CENSORED

Figure 2: NBS students entering their information in the NBS ABC Excel Spreadsheet

The NBS Accountancy and Business Club (ABC) Add/Drop Excel spreadsheet is currently the primary mode for NBS students to conduct their module swapping. The spreadsheet is separated into different categories[3] that makes it easier for students in the same group to search for the relevant course and index that they need. The spreadsheet is generally received positively by NBS students.

Despite its popularity, we have noticed that the spreadsheet has also allowed students to take a passive approach, as they enter their own particulars and expect to be contacted by others who are viewing the spreadsheet. A probable swap may not happen as a result. The spreadsheet has its disadvantages on both schools with large intake and schools with

[2] Inconsistencies in user inputs may arise when a user tries to find the course by its course number (eg. CZ1011) but other users may have written information like course name (eg. in the same case: Engineering Math 1, E Math I etc.)
[3] Categories include separating between Year 1 Group A and Year 1 Group B students, as well as separate sheets catering to the Accountancy and different specialisations in the Business course form Year 2 onwards.

small intake. For schools with large intake (like NBS), the spreadsheet is very extensive, which makes searching for the right pair of complementary data significantly harder. On the other hand, for schools with smaller intake[4] (like School of Medicine), the low number of user inputs may lead to not having enough information in the spreadsheet to allow students to find a suitable complementary data.

Significance of our problem

We went around surveying the students of NTU, asking them to rate the difficulty of finding someone to swap their module with on a scale of 1 to 10 with 1 being the easiest, the average number of people they had to find to swap their index, as well as the duration they took to swap their index. We found that generally, the difficulty is rated highly. Furthermore, we found that those who rated slightly easier were from faculties which has tools such as, a mass WhatsApp group in the faculty of SPMS and the ABC Add/Drop Spreadsheet in the faculty of NBS, to assist students to swap the course indexes. In conclusion, we feel that the current tools are inadequate and ineffective in facilitating the swapping of course indexes between students as explained in above sections.
Our survey data[5], coupled with our analysis of current tools available, all clearly points to a need for a more efficient tool for the students of NTU for swapping of course indexes. We hope to be able to achieve this with the creation of a new, interactive, and most importantly, effective platform to facilitate the swapping of course indexes between NTU students across all faculties.

Our proposed solution

Team Scrubzbot has created a Telegram bot to aid NTU students in swapping their course index efficiently. The NTU Mod Swap bot[6] (We will refer it as "Our Telegram bot" for the rest of the report) is designed to be an efficient and user-friendly tool for NTU students to facilitate their index swapping. Figure 16[7] is a screenshot of our Telegram bot. Our Telegram bot has four main features:

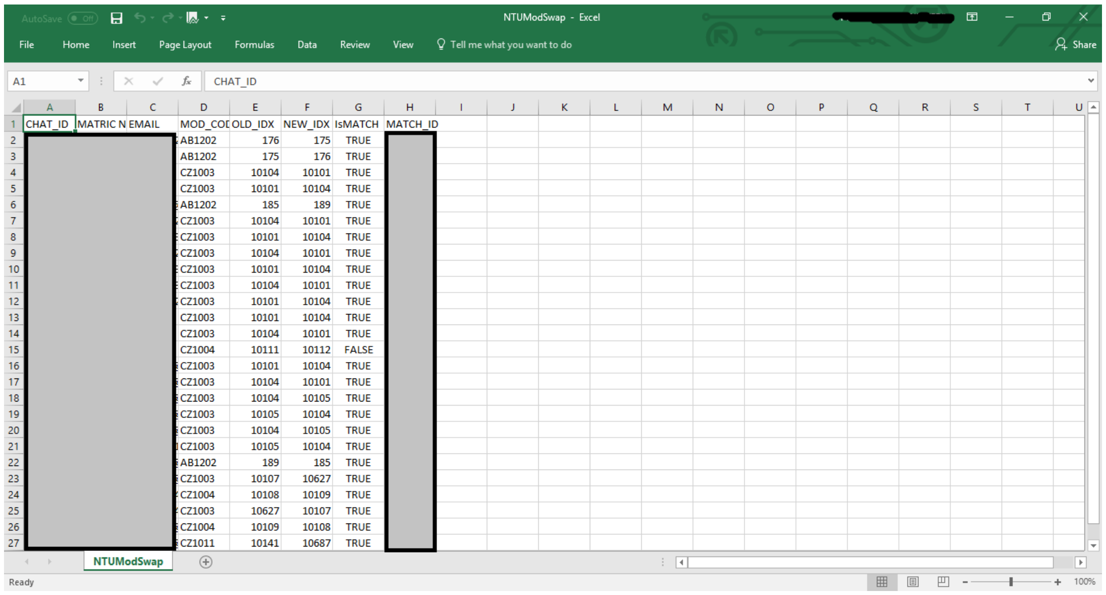
1. Check various indexes for the course

It is necessary, for students to know the timings of their current index and the other indexes that they can be swapped to. It would not be intuitive for students to go to NTU's website just to find the timings for the respective course indexes. Furthermore, we realised that there are students who are not aware of the NTU website that displays all the indexes for all the courses. We have decided to do web scraping from the NTU website and integrate the display of course indexes into our Telegram bot. By entering the valid course code, the student will view all the course indexes for the course, as shown in Figure 17.

[4] As there are no other schools that has used the spreadsheet other than NBS, we can only gather roadside opinion and predict the usability of the spreadsheet for the other schools.
[5] Our survey data can be found in the Appendix, from Figures 11 to 15, under "Survey results".
[6] We will refer to it as "our Telegram bot" for the rest of the report.
[7] The screenshots of the main functions of the Telegram bot are included in the Appendix, from Figures 16 to 22.

2. Input entry to swap course index:

This is the most important feature in our Telegram bot, which allows students to enter their details to swap their course index. The student will have to input the following details (in order when prompted): Valid NTU matriculation number → Valid NTU email → Valid NTU course code → Current index that the user has → The user's preferred index for the course. Each valid entry will be stored into our csv file, as shown in Figure 3. The student may enter more entries as they wish should they want to swap index for more courses or indicate preferences for multiple indexes in the same course.



	A	B	C	D	E	F	G	H
1	CHAT_ID	MATRIC N	EMAIL	MOD_COD	OLD_IDX	NEW_IDX	IsMATCH	MATCH_ID
2	[redacted]	[redacted]	[redacted]	AB1202	176	175	TRUE	[redacted]
3	[redacted]	[redacted]	[redacted]	AB1202	175	176	TRUE	[redacted]
4	[redacted]	[redacted]	[redacted]	CZ1003	10104	10101	TRUE	[redacted]
5	[redacted]	[redacted]	[redacted]	CZ1003	10101	10104	TRUE	[redacted]
6	[redacted]	[redacted]	[redacted]	AB1202	185	189	TRUE	[redacted]
7	[redacted]	[redacted]	[redacted]	CZ1003	10104	10101	TRUE	[redacted]
8	[redacted]	[redacted]	[redacted]	CZ1003	10101	10104	TRUE	[redacted]
9	[redacted]	[redacted]	[redacted]	CZ1003	10104	10101	TRUE	[redacted]
10	[redacted]	[redacted]	[redacted]	CZ1003	10101	10104	TRUE	[redacted]
11	[redacted]	[redacted]	[redacted]	CZ1003	10104	10101	TRUE	[redacted]
12	[redacted]	[redacted]	[redacted]	CZ1003	10101	10104	TRUE	[redacted]
13	[redacted]	[redacted]	[redacted]	CZ1003	10101	10104	TRUE	[redacted]
14	[redacted]	[redacted]	[redacted]	CZ1003	10104	10101	TRUE	[redacted]
15	[redacted]	[redacted]	[redacted]	CZ1004	10111	10112	FALSE	[redacted]
16	[redacted]	[redacted]	[redacted]	CZ1003	10101	10104	TRUE	[redacted]
17	[redacted]	[redacted]	[redacted]	CZ1003	10104	10101	TRUE	[redacted]
18	[redacted]	[redacted]	[redacted]	CZ1003	10104	10105	TRUE	[redacted]
19	[redacted]	[redacted]	[redacted]	CZ1003	10105	10104	TRUE	[redacted]
20	[redacted]	[redacted]	[redacted]	CZ1003	10104	10105	TRUE	[redacted]
21	[redacted]	[redacted]	[redacted]	CZ1003	10105	10104	TRUE	[redacted]
22	[redacted]	[redacted]	[redacted]	AB1202	189	185	TRUE	[redacted]
23	[redacted]	[redacted]	[redacted]	CZ1003	10107	10627	TRUE	[redacted]
24	[redacted]	[redacted]	[redacted]	CZ1004	10108	10109	TRUE	[redacted]
25	[redacted]	[redacted]	[redacted]	CZ1003	10627	10107	TRUE	[redacted]
26	[redacted]	[redacted]	[redacted]	CZ1004	10109	10108	TRUE	[redacted]
27	[redacted]	[redacted]	[redacted]	CZ1011	10141	10687	TRUE	[redacted]

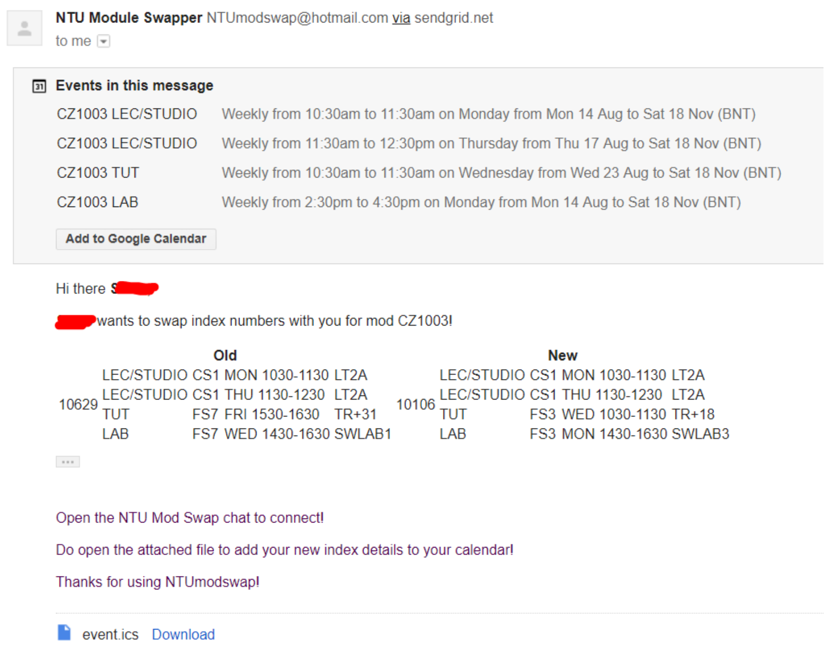
Figure 3: Screenshot of csv

We have used the inline keyboard to allow users to select the current course index that they have, as shown in Figure 18(a). In addition to showing all course indexes, there is an additional option for user to select to view all the course index timings[8], as shown in Figure 18(b). After choosing their current index, the user will be brought to a second inline keyboard with all the course indexes except for the course index that they previously chose as their current index, as shown in Figure 18(c). This is essential in the logic flow of our program as we expect that users should not choose their current index as their preferred one, which would null the fundamental purpose of swapping course index.

We have implemented email validation to ensure that users have entered a valid email (preferably their NTU email), so that they can receive an email notification once there is a successful swap in their course index. Upon entering their email, the server email validation will run in the background. After all the other entries are entered, our Telegram bot will retrieve the API response and return an appropriate response to the user. Figures 19(a) and 19(b) show the response in the event of a valid and an invalid email respectively, the latter rendering the whole data entry invalid.

[8] This is like feature 1 where students can view all the course's indexes and timings. We have implemented it in feature 2 as well to allow users to check and remember the course's indexes and timings right before they choose their current and preferred index.

If our database receives a pair of complementary data, it will send a notification to both users' email to notify them of the swap, as well as an iCalendar file containing the new index details attached to the email. as shown in Figure 4. In addition, both users will also receive a notification from our Telegram bot, informing them of the swap as well as the other student's Telegram user ID, as shown in Figure 20. The latter allows the student to contact the other student as their username is made available.



NTU Module Swapper NTUmodswap@hotmail.com via sendgrid.net
to me

Events in this message

CZ1003 LEC/STUDIO	Weekly from 10:30am to 11:30am on Monday from Mon 14 Aug to Sat 18 Nov (BNT)
CZ1003 LEC/STUDIO	Weekly from 11:30am to 12:30pm on Thursday from Thu 17 Aug to Sat 18 Nov (BNT)
CZ1003 TUT	Weekly from 10:30am to 11:30am on Wednesday from Wed 23 Aug to Sat 18 Nov (BNT)
CZ1003 LAB	Weekly from 2:30pm to 4:30pm on Monday from Mon 14 Aug to Sat 18 Nov (BNT)

Add to Google Calendar

Hi there [redacted]

[redacted] wants to swap index numbers with you for mod CZ1003!

	Old						New				
10629	LEC/STUDIO	CS1	MON	1030-1130	LT2A	10106	LEC/STUDIO	CS1	MON	1030-1130	LT2A
	LEC/STUDIO	CS1	THU	1130-1230	LT2A		LEC/STUDIO	CS1	THU	1130-1230	LT2A
	TUT	FS7	FRI	1530-1630	TR+31		TUT	FS3	WED	1030-1130	TR+18
	LAB	FS7	WED	1430-1630	SWLAB1		LAB	FS3	MON	1430-1630	SWLAB3

Open the NTU Mod Swap chat to connect!

Do open the attached file to add your new index details to your calendar!

Thanks for using NTUmodswap!

event.ics Download

Figure 4: Email sent to user after a successful index swap.

3. Check pending status in index swapping

This feature allows students to check their pending entries. This comes in handy for students who have multiple entries to keep track of their entries. The index swapping status shows the normal details of course code, current index and preferred index of the user. In addition, it shows the waiting list queue where the user's entry is at for the course, giving the student a good gauge on his priority in getting a swap for his course index. An example of this feature is shown in Figure 21.

4. Cancel pending status in index swapping

In addition to the previous feature of allowing students to see their current pending entries, they may also cancel any of their entries as they wish. From their list of current entries, they may choose the number associated with the entry to be cancelled. Our Telegram bot will show the removal of the entry and the remaining entries for the user, as well as output a worded notification to inform users that the entry has indeed been successfully removed, as shown in Figure 22.

Technical details on how our Telegram bot works can be found in Figures 23 and 24.

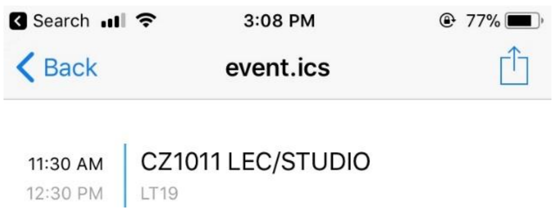
Service Integration

1. Web Scraping

Our primary data source was the NTU class schedules portal, from which we scraped and process the required content via urllib and Beautiful Soup. We had to closely look at the page structure to sieve out the exact elements required, which we subsequently loaded into our own data structures.

2. Email API (Sending)

As an additional notification method, we integrated e-mail messaging. The intent behind this is multifold - for redundancy, clarity as well as to facilitate calendar integration, which is another key selling point of our product. Expanding the last point, one limitation of iOS is that .ics files received via Telegram are treated as “attachments” and not as importable calendar files.



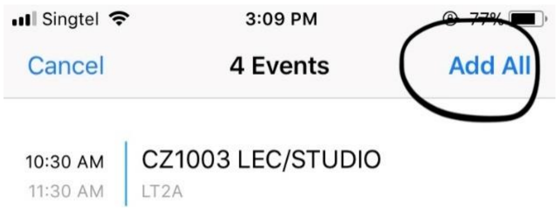


Figure 5: iCalendar file open in Telegram   Figure 6. iCalendar file open in email client

Hence, we leveraged SaaS platform Sendgrid, taking advantage of their Python library to send rich HTML messages with complete information and an attached iCalendar file, which would be recognized correctly by iOS. Content aside, using Sendgrid allowed us to not have to worry about the unreliability of self-hosted solutions such as PHP mail() or the hassle of configuring SMTP.

On the point of redundancy, we cannot forget that some states routinely block messaging apps such as Telegram, as well as the fact that Telegram is susceptible to downtime. Hence, as integrated as we want our solution to be, we feel that this feature would be useful in such an event - if anything, it would go some way in reducing the anxiety of the counterparty, who would be faced with an unresponsive ‘swapper’ in an ever-changing slot availability situation.

3. Email API (Validation)

To conserve resources and weed out spam, we also went one step further by leveraging external APIs to validate users’ email addresses. Using the Python library provided by Neverbounce, we incorporated a check prior to database insertion, ending the user session if the email was confirmed to be invalid or undeliverable.

Apart from the emails being syntactically valid, we also wanted to ensure that the user can receive the email notification and fully leverage the features of our bot. Neverbounce’s ability to check whether the domain was connectable and had a valid MX record, among other features, allowed us to do that.

How is our idea better than current tools available?

The most crucial merit to our Telegram bot is the presence of automation. One of the main concerns we have with the existing index swapping tools is the use of mere display of information. This forces users to source out for a partner to swap their course index with. With our NTU Mod Swap bot, the bot can check its database for complementary data, and conduct the swap for the students. The students do not need to find another student on their own to check if they have the index that the student wanted. Overall, the efficiency of index swapping has increased.

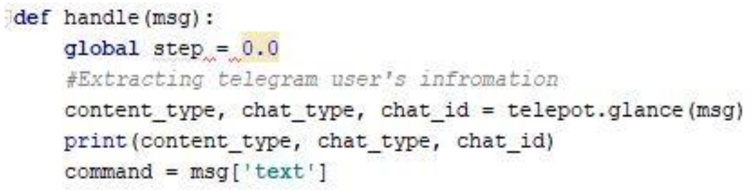
One of the student users has provided a positive feedback about our Telegram bot. *"[The Telegram bot] saves us a lot of hassle to find another person to swap index with… It is very mobile-friendly and interactive. Overall, I would fully support this innovative idea of building a Telegram bot to help us swap (our course) index."* - James Yap Junwei.

Beyond index swapping, we go one step further to provide a one-click method for the user to load his updated schedule into his personal calendar. ICalendar is a commonly used method of exchanging event information in the school, as evidenced from the incorporation of iCalendar to the school's Event Registration System as well as the RSVP system for the school's career portal, CareerAxis. Students would therefore be familiar with the workflow and this value-added feature would enhance the post-swap user experience as the updated information would be reflected in the user's personal calendar with minimal effort. A member from another project group, who prefers to remain unnamed, mentioned that the motivation behind their telegram bot was "*missing class after class*" - with this solution, swaps and changes notwithstanding, such scenarios would be a thing of the past.

Problems we faced during the project

1. Unit counter problem

When we tested the bot initially, we have assigned a global step count to each step (Step 2.0 is to start the bot, step 2.1 is for the user to select the first feature to view all course indexes etc.). However, the step count will continue running while a user is still using the bot. This poses a problem to allowing multiple users to access the bot at the same time, as well as another user having unauthorised access to the previous user's input as they are still using the bot[9].

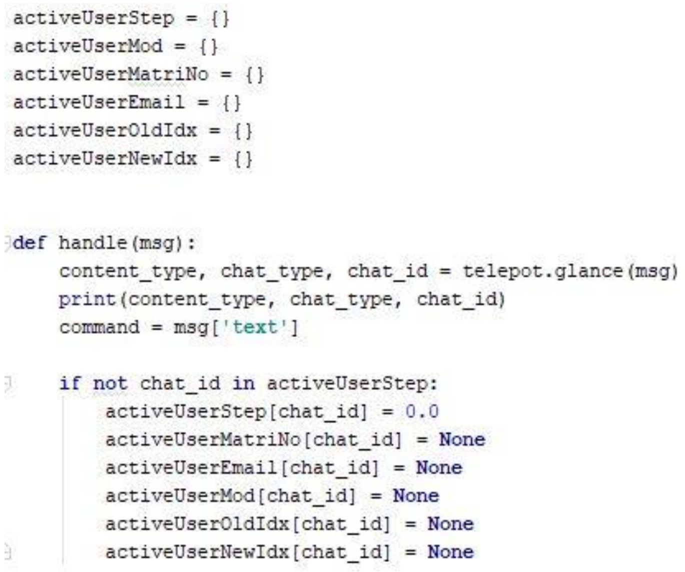


```
def handle(msg):
    global step = 0.0
    #Extracting telegram user's infromation
    content_type, chat_type, chat_id = telepot.glance(msg)
    print(content_type, chat_type, chat_id)
    command = msg['text']
```

Figure 7. Code for global step counter (resulting in unit counter problem)

[9] For example, user A is using our Telegram bot at the step where he wants to input the course that he wants to swap his index for. However, when user B wants to use our Telegram bot, he will find himself at that step (of inputting the course code to swap index for) rather than starting the bot, the step where he is supposed to be at.

We have subsequently modified our code to include a unit counter feature. This helps to assign a step to each unique user as they are using the bot. Using the same example, while the unit counter has assigned user A to step 2.1, user B would be uniquely assigned to be at step 2.0 when he starts the bot.



```
activeUserStep = {}
activeUserMod = {}
activeUserMatriNo = {}
activeUserEmail = {}
activeUserOldIdx = {}
activeUserNewIdx = {}


def handle(msg):
    content_type, chat_type, chat_id = telepot.glance(msg)
    print(content_type, chat_type, chat_id)
    command = msg['text']

    if not chat_id in activeUserStep:
        activeUserStep[chat_id] = 0.0
        activeUserMatriNo[chat_id] = None
        activeUserEmail[chat_id] = None
        activeUserMod[chat_id] = None
        activeUserOldIdx[chat_id] = None
        activeUserNewIdx[chat_id] = None
```

Figure 8. Code for activeUserStep counter

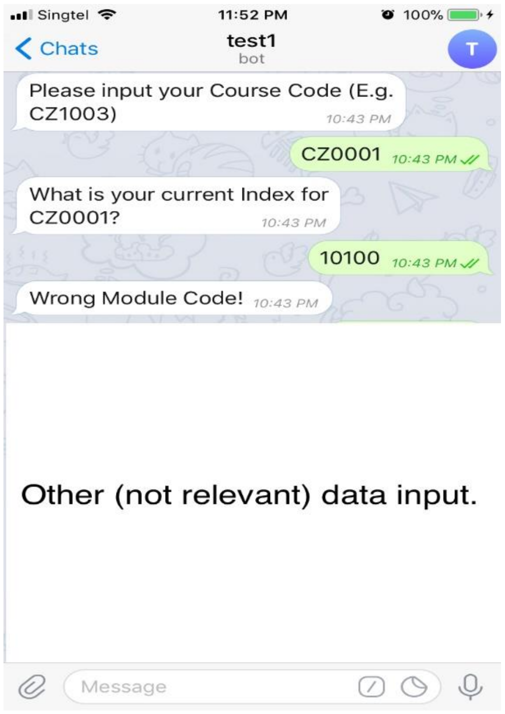
2. Waiting time for server email validation

After implementing the server email validation feature, we realised that the service had to have adequate time to run to produce meaningful responses. This time was observed to be around 5 seconds, during which the bot would not respond to any user as the code was initially blocking. We realised that this waiting time was not only inefficient on the user's side, but also had the spill over effect of jamming other users' sessions as well regardless of whatever step they were on. Despite this, we felt that it was still necessary for the email validation to process and produce a meaningful response, as that would ensure the integrity of our data.

To increase user-friendliness in using our Telegram bot, we allowed the server email validation to run as a separate thread in the background, initiated when the user inputs an email.
This allows the bot to continue processing incoming commands from all users. As for the user whose email is being validated, we allow him to continue entering his course code, current index and new index, to maintain the momentum. We then check on the server response just before database insertion. This allows us to achieve both goals of ensuring the data given is valid while not interrupting the index swapping flow.

3. Logic problem for courses with one index only

When we conducted our UAT, one of the students have tried to input the course code CZ0001 - Engineers and Society, which only has one course index[10]. Upon putting the course code, there were significant glitches. First, there was no inline keyboard input for the sole index available. The bot also prompted the user of "Wrong input" even though the user has entered the valid sole course index into the bot, as shown in Figure 9. Logically, students would not want to make an index swap for such courses as there is only one index available for everyone, with no other options to swap. We have implemented a separate response for courses with a sole index. Instead of simply showing the residual response "Wrong input!", the bot will inform them to not input for such courses, as shown in Figure 10.



Singtel 11:52 PM 100%

Chats test1 bot

Please input your Course Code (E.g. CZ1003) 10:43 PM

CZ0001 10:43 PM

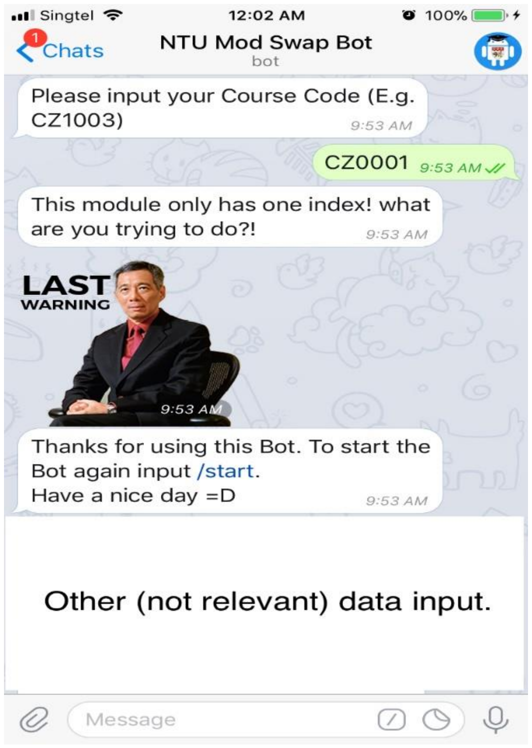
What is your current Index for CZ0001? 10:43 PM

10100 10:43 PM

Wrong Module Code! 10:43 PM

Other (not relevant) data input.

Message



Singtel 12:02 AM 100%

Chats NTU Mod Swap Bot bot

Please input your Course Code (E.g. CZ1003) 9:53 AM

CZ0001 9:53 AM

This module only has one index! what are you trying to do?! 9:53 AM

9:53 AM

Thanks for using this Bot. To start the Bot again input /start.
Have a nice day =D 9:53 AM

Other (not relevant) data input.

Message

Figure 9. Before - input single course index   Figure 10. After – input single course index

[10] This course is solely lecture-based, resulting in all students who are supposed to attend the common lecture having the same course index. A similar problem is likely to arise in online courses where all students have the same access to the online modules.

Conclusion/Moving On

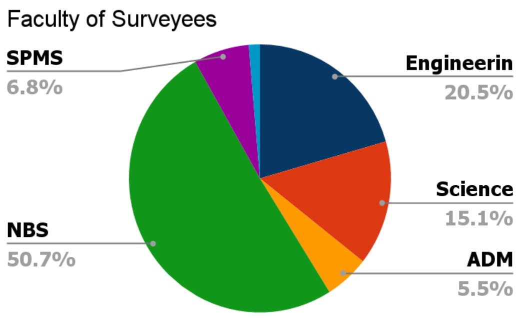
Due to the small scale of our project, we acknowledge that we have some room of improvement that we have not yet been implemented but can hopefully be implemented in our Telegram bot to make it sustainable in the future.

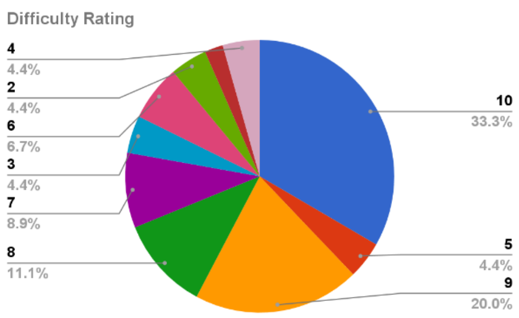
We have considered to introduce filters and constraints for the students when want to check their course index timings or when they are selecting their preferred index. We believe that this feature will be very useful to assist students by showing the course index timings available for the student, according to the constraints that they have set. It is also more user-oriented than solely displaying all the course indexes and timings. If our project can be done in a long term and with a larger bandwidth of technical knowledge, this would be a feature that our group would want to work on for the users.

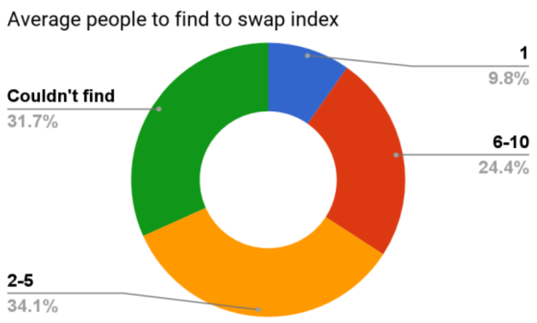
We have also considered that not all NTU students have a Telegram account. In a long run, we hope to be able to expand our project to help the mass of NTU students. We could consider having complementary web services that inherits the efficiency and user-friendliness of our Telegram bot, to cater to the mass of NTU students who do not have a Telegram account. Combining the databases of our Telegram bot and its complimentary website, this can make our index swapping tools a popular option for the mass of NTU students.

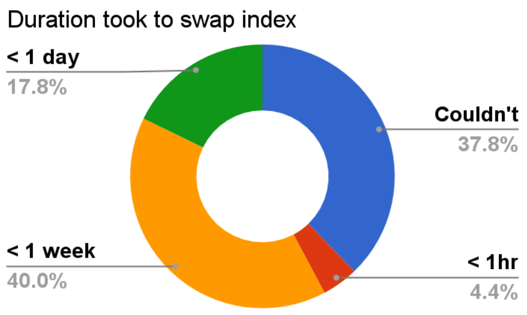
Appendix

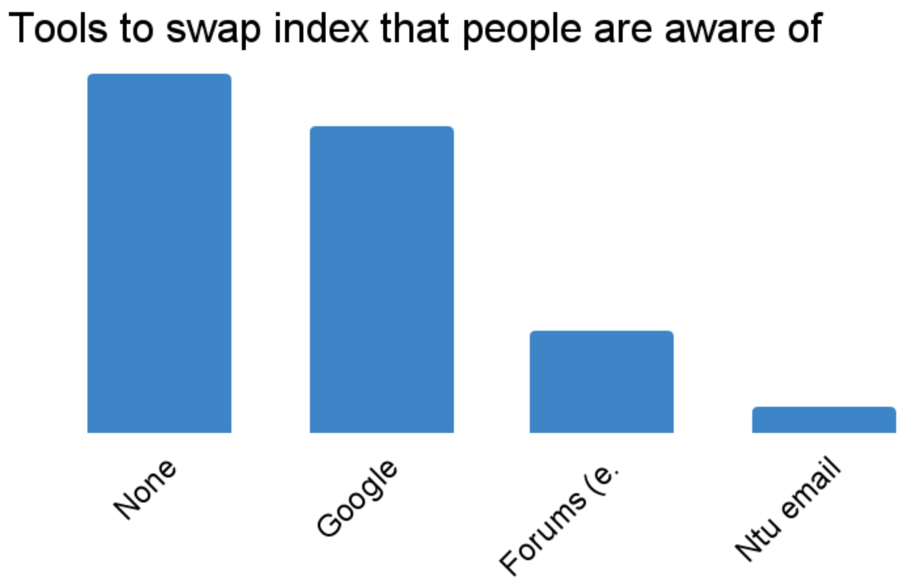
Survey results





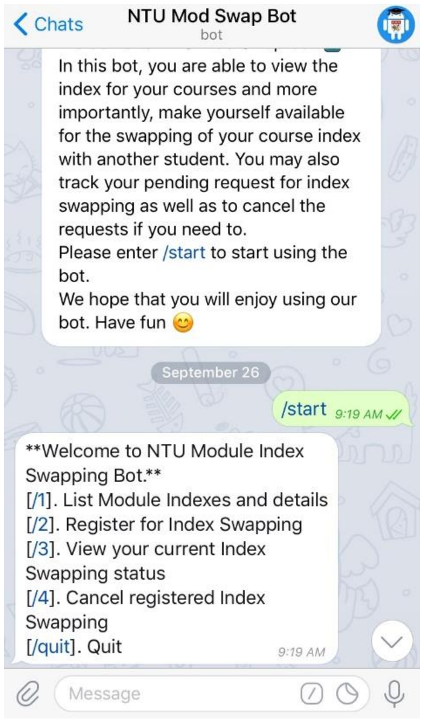






Figures 11 - 15: survey results
Survey link: https://goo.gl/forms/REwTMA1OSNKSBtcE2

Screenshots of our Telegram bot


Chats
NTU Mod Swap Bot
bot

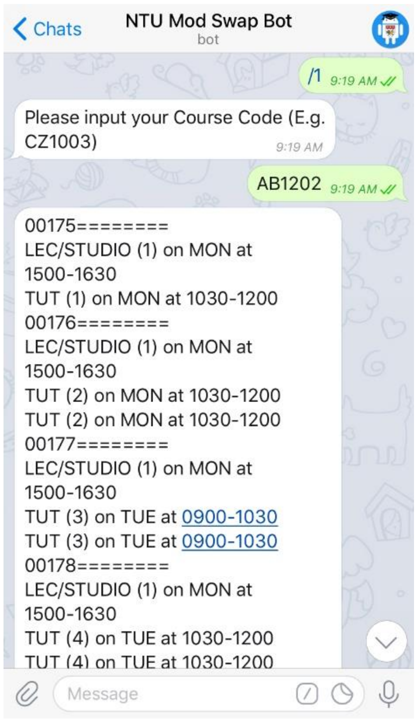
In this bot, you are able to view the index for your courses and more importantly, make yourself available for the swapping of your course index with another student. You may also track your pending request for index swapping as well as to cancel the requests if you need to.
Please enter /start to start using the bot.
We hope that you will enjoy using our bot. Have fun 😊

September 26

/start 9:19 AM

Welcome to NTU Module Index Swapping Bot.
[/1]. List Module Indexes and details
[/2]. Register for Index Swapping
[/3]. View your current Index Swapping status
[/4]. Cancel registered Index Swapping
[/quit]. Quit 9:19 AM

Figure 16: Introduction screen


Chats
NTU Mod Swap Bot
bot

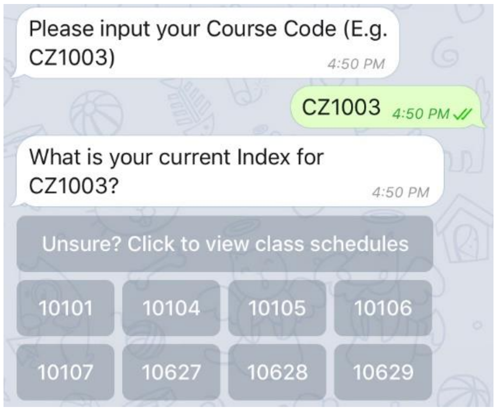
/1 9:19 AM

Please input your Course Code (E.g. CZ1003) 9:19 AM

AB1202 9:19 AM

```
00175========
LEC/STUDIO (1) on MON at
1500-1630
TUT (1) on MON at 1030-1200
00176========
LEC/STUDIO (1) on MON at
1500-1630
TUT (2) on MON at 1030-1200
TUT (2) on MON at 1030-1200
00177========
LEC/STUDIO (1) on MON at
1500-1630
TUT (3) on TUE at 0900-1030
TUT (3) on TUE at 0900-1030
00178========
LEC/STUDIO (1) on MON at
1500-1630
TUT (4) on TUE at 1030-1200
TUT (4) on TUE at 1030-1200
```

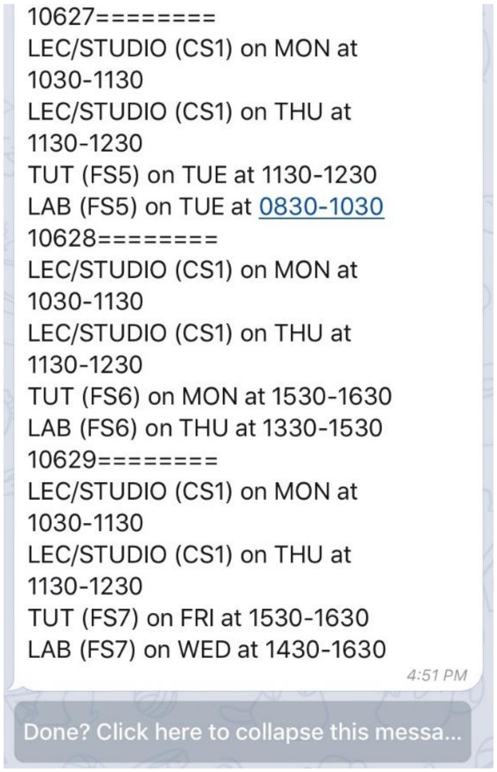
Figure 17: Feature 1 - Check course indexes


Please input your Course Code (E.g. CZ1003) 4:50 PM

CZ1003 4:50 PM

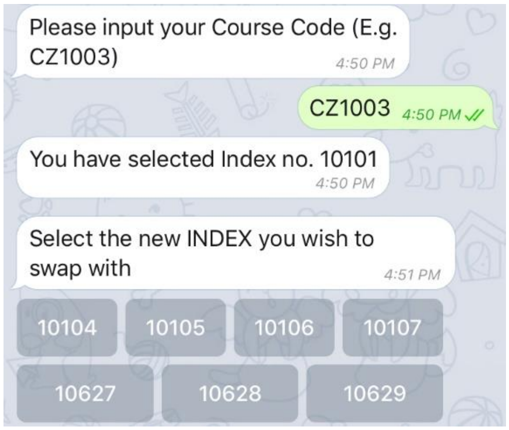
What is your current Index for CZ1003? 4:50 PM

Figure 18(a): Current index keyboard



```
10627========
LEC/STUDIO (CS1) on MON at
1030-1130
LEC/STUDIO (CS1) on THU at
1130-1230
TUT (FS5) on TUE at 1130-1230
LAB (FS5) on TUE at 0830-1030
10628========
LEC/STUDIO (CS1) on MON at
1030-1130
LEC/STUDIO (CS1) on THU at
1130-1230
TUT (FS6) on MON at 1530-1630
LAB (FS6) on THU at 1330-1530
10629========
LEC/STUDIO (CS1) on MON at
1030-1130
LEC/STUDIO (CS1) on THU at
1130-1230
TUT (FS7) on FRI at 1530-1630
LAB (FS7) on WED at 1430-1630
4:51 PM
```

Figure 18(b): Viewing course index after selecting “Unsure” in the inline keyboard.



Please input your Course Code (E.g. CZ1003) 4:50 PM

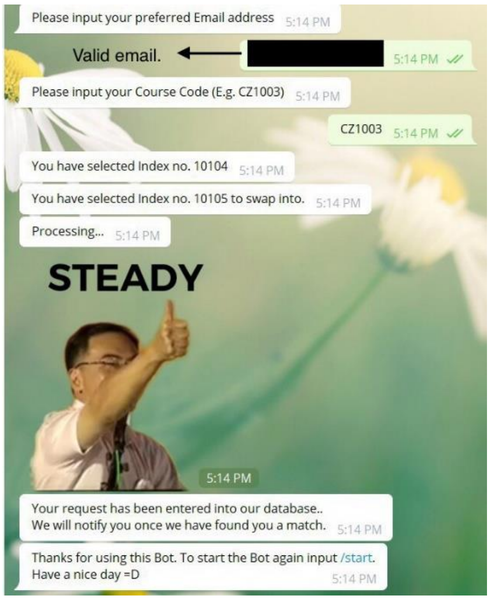
CZ1003 4:50 PM

You have selected Index no. 10101 4:50 PM

Select the new INDEX you wish to swap with 4:51 PM

Figure 18(c): new index keyboard

19(a):



Please input your preferred Email address 5:14 PM

[redacted] 5:14 PM

Please input your Course Code (E.g. CZ1003) 5:14 PM

CZ1003 5:14 PM

You have selected Index no. 10104 5:14 PM

You have selected Index no. 10105 to swap into. 5:14 PM

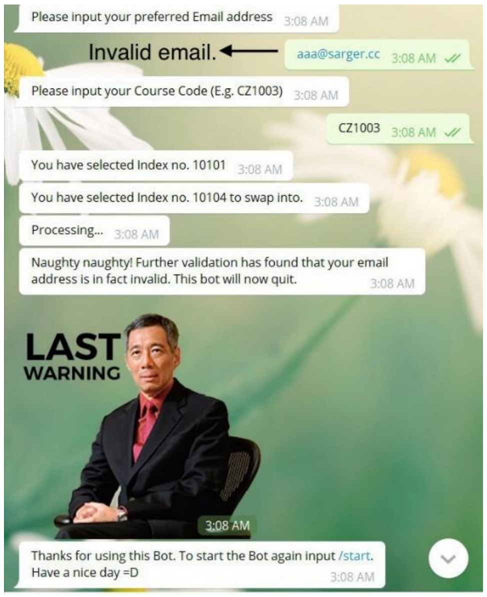
Processing... 5:14 PM

5:14 PM

Your request has been entered into our database..
We will notify you once we have found you a match. 5:14 PM

Thanks for using this Bot. To start the Bot again input /start.
Have a nice day =D 5:14 PM

19(b):



Please input your preferred Email address 3:08 AM

aaa@sarger.cc 3:08 AM

Please input your Course Code (E.g. CZ1003) 3:08 AM

CZ1003 3:08 AM

You have selected Index no. 10101 3:08 AM

You have selected Index no. 10104 to swap into. 3:08 AM

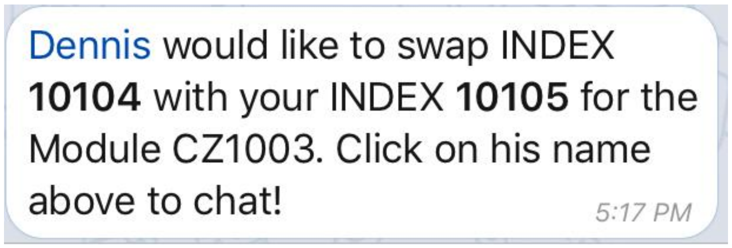
Processing... 3:08 AM

Naughty naughty! Further validation has found that your email address is in fact invalid. This bot will now quit. 3:08 AM

3:08 AM

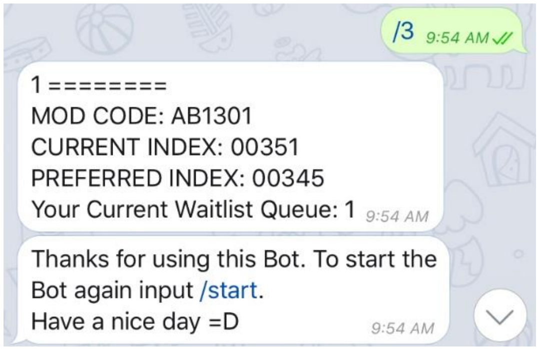
Thanks for using this Bot. To start the Bot again input /start.
Have a nice day =D 3:08 AM

Figure 19: email validation -
(a) Valid email.
(b) Invalid email.



Dennis would like to swap INDEX **10104** with your INDEX **10105** for the Module CZ1003. Click on his name above to chat! 5:17 PM

Figure 20: Telegram notification to user to indicate successful swap.

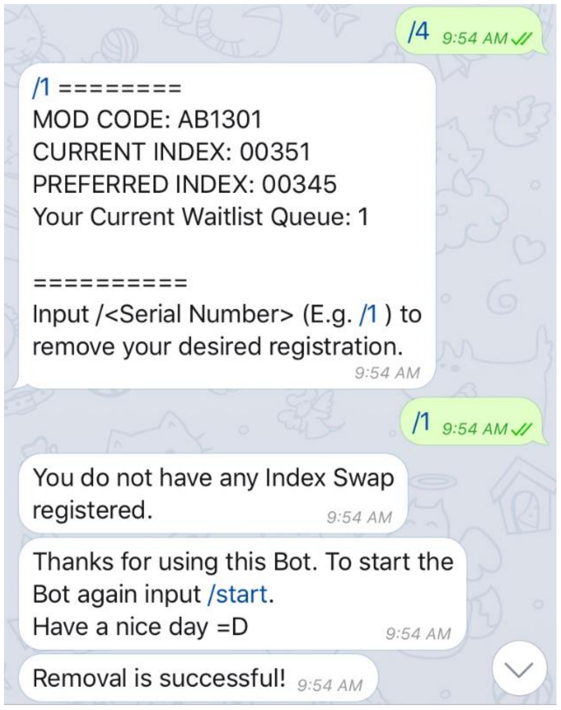


/3 9:54 AM

```
1 ========
MOD CODE: AB1301
CURRENT INDEX: 00351
PREFERRED INDEX: 00345
Your Current Waitlist Queue: 1
```
9:54 AM

Thanks for using this Bot. To start the Bot again input /start.
Have a nice day =D 9:54 AM

Figure 21: Feature 3 - Check current pending entries



```
/4 9:54 AM

/1 ========
MOD CODE: AB1301
CURRENT INDEX: 00351
PREFERRED INDEX: 00345
Your Current Waitlist Queue: 1

==========
Input /<Serial Number> (E.g. /1 ) to
remove your desired registration.
9:54 AM

/1 9:54 AM

You do not have any Index Swap
registered. 9:54 AM

Thanks for using this Bot. To start the
Bot again input /start.
Have a nice day =D 9:54 AM

Removal is successful! 9:54 AM
```

Figure 22: Feature 4 - Option to cancel pending entries

Flowchart

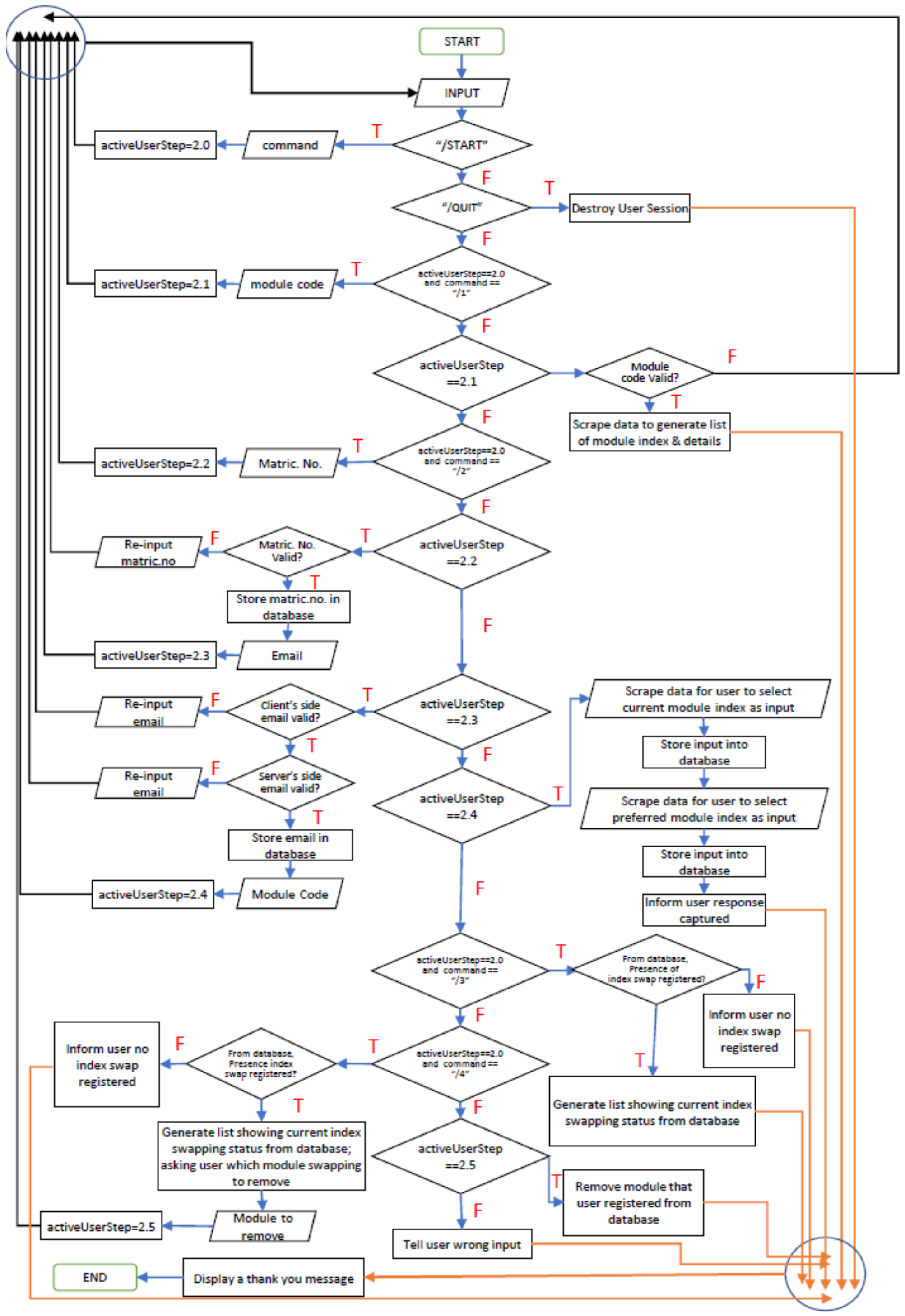

START
INPUT
"/START"
T
command
activeUserStep=2.0
F
"/QUIT"
T
Destroy User Session
F
activeUserStep==2.0 and command == "/1"
T
module code
activeUserStep=2.1
F
activeUserStep ==2.1
Module code Valid?
F
T
Scrape data to generate list of module index & details
F
activeUserStep==2.0 and command == "/2"
T
Matric. No.
activeUserStep=2.2
F
activeUserStep ==2.2
T
Matric. No. Valid?
F
Re-input matric.no
T
Store matric.no. in database
Email
activeUserStep=2.3
F
activeUserStep ==2.3
T
Client's side email valid?
F
Re-input email
T
Server's side email valid?
F
Re-input email
T
Store email in database
Module Code
activeUserStep=2.4
F
activeUserStep ==2.4
T
Scrape data for user to select current module index as input
Store input into database
Scrape data for user to select preferred module index as input
Store input into database
Inform user response captured
F
activeUserStep==2.0 and command == "/3"
T
From database, Presence of index swap registered?
F
Inform user no index swap registered
T
Generate list showing current index swapping status from database
F
activeUserStep==2.0 and command == "/4"
T
From database, Presence index swap registered?
F
Inform user no index swap registered
T
Generate list showing current index swapping status from database; asking user which module swapping to remove
Module to remove
activeUserStep=2.5
F
activeUserStep ==2.5
T
Remove module that user registered from database
F
Tell user wrong input
Display a thank you message
END

Figure 23: Low level flow chart

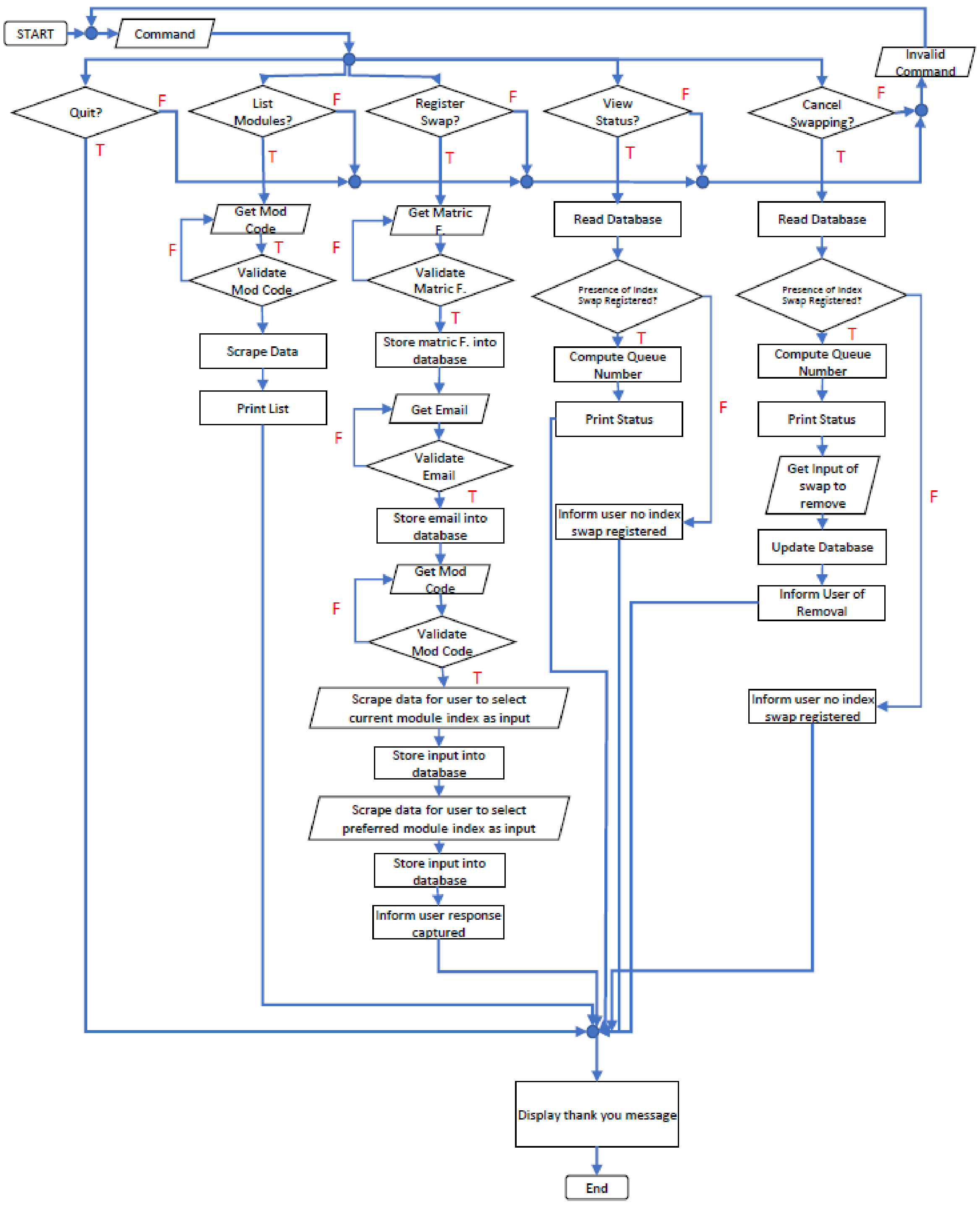
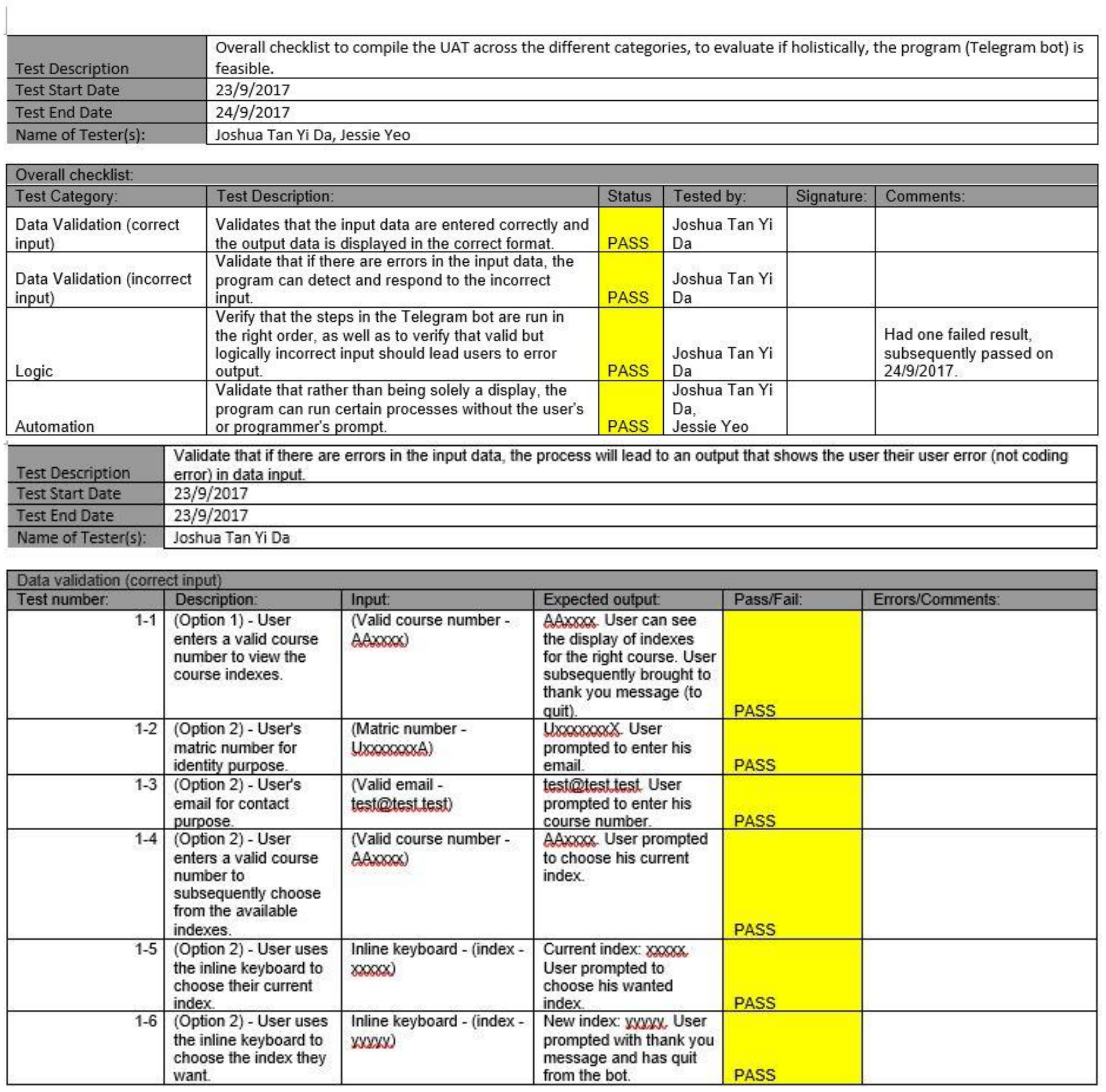

START
Command
Invalid Command
Quit?
List Modules?
Register Swap?
View Status?
Cancel Swapping?
T
F
Get Mod Code
Validate Mod Code
Scrape Data
Print List
Get Matric F.
Validate Matric F.
Store matric F. into database
Get Email
Validate Email
Store email into database
Get Mod Code
Validate Mod Code
Scrape data for user to select current module index as input
Store input into database
Scrape data for user to select preferred module index as input
Store input into database
Inform user response captured
Read Database
Presence of Index Swap Registered?
Compute Queue Number
Print Status
Inform user no index swap registered
Read Database
Presence of Index Swap Registered?
Compute Queue Number
Print Status
Get Input of swap to remove
Update Database
Inform User of Removal
Inform user no index swap registered
Display thank you message
End

Figure 24: High level flowchart

User Acceptance Testing

UAT

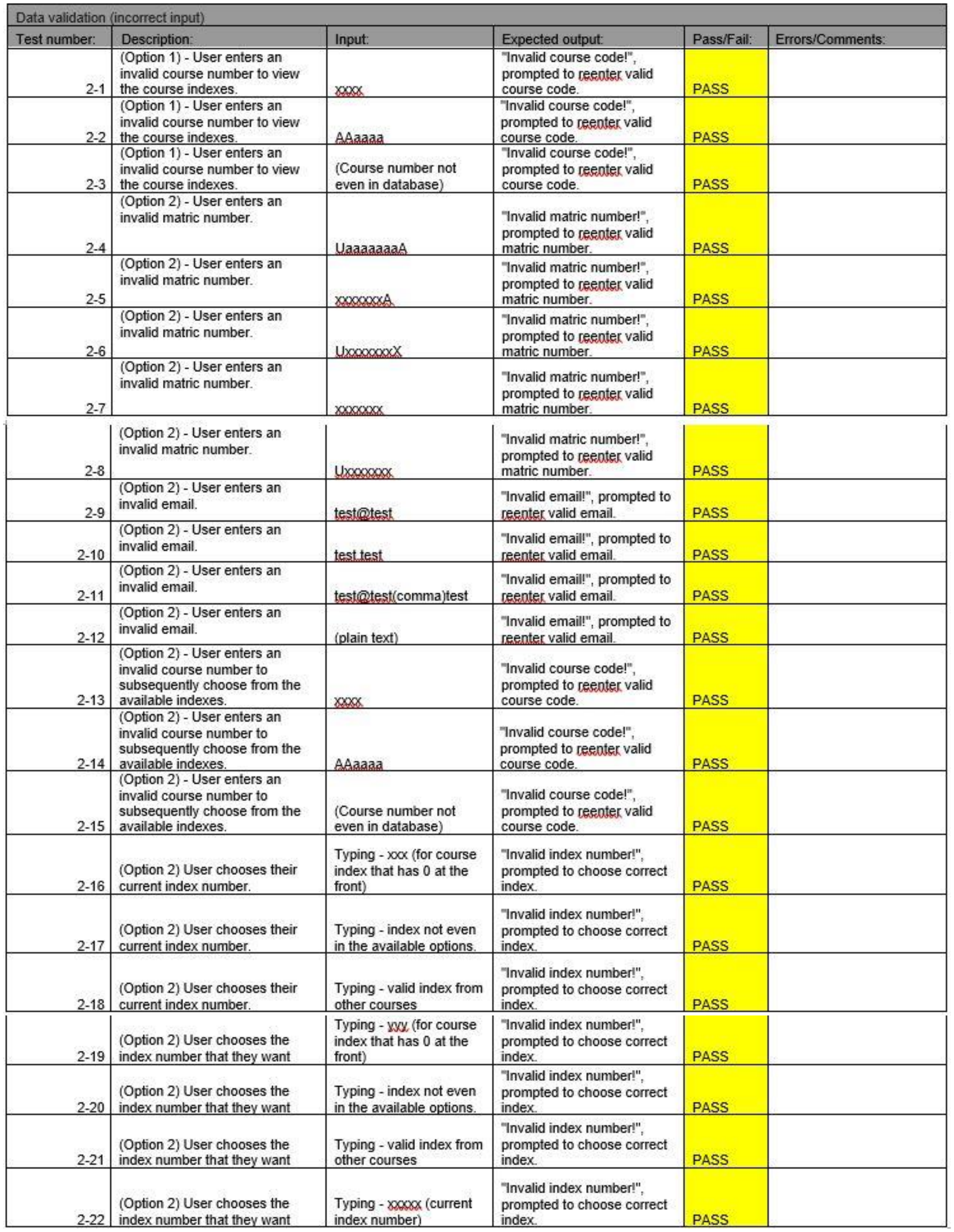


Test Description	Overall checklist to compile the UAT across the different categories, to evaluate if holistically, the program (Telegram bot) is feasible.
Test Start Date	23/9/2017
Test End Date	24/9/2017
Name of Tester(s):	Joshua Tan Yi Da, Jessie Yeo

Overall checklist:					
Test Category:	Test Description:	Status	Tested by:	Signature:	Comments:
Data Validation (correct input)	Validates that the input data are entered correctly and the output data is displayed in the correct format.	PASS	Joshua Tan Yi Da		
Data Validation (incorrect input)	Validate that if there are errors in the input data, the program can detect and respond to the incorrect input.	PASS	Joshua Tan Yi Da		
Logic	Verify that the steps in the Telegram bot are run in the right order, as well as to verify that valid but logically incorrect input should lead users to error output.	PASS	Joshua Tan Yi Da		Had one failed result, subsequently passed on 24/9/2017.
Automation	Validate that rather than being solely a display, the program can run certain processes without the user's or programmer's prompt.	PASS	Joshua Tan Yi Da, Jessie Yeo		

Test Description	Validate that if there are errors in the input data, the process will lead to an output that shows the user their user error (not coding error) in data input.
Test Start Date	23/9/2017
Test End Date	23/9/2017
Name of Tester(s):	Joshua Tan Yi Da

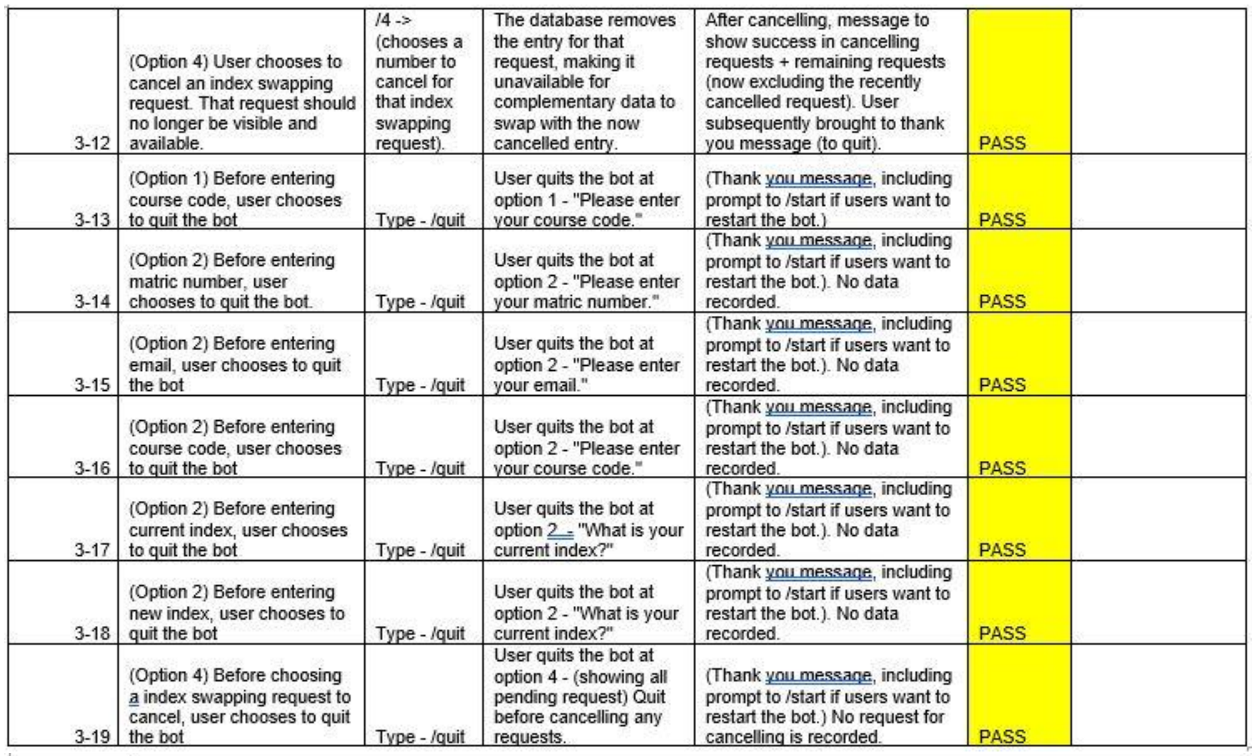
Data validation (correct input)					
Test number:	Description:	Input:	Expected output:	Pass/Fail:	Errors/Comments:
1-1	(Option 1) - User enters a valid course number to view the course indexes.	(Valid course number - AAxxxx)	AAxxxx. User can see the display of indexes for the right course. User subsequently brought to thank you message (to quit).	PASS	
1-2	(Option 2) - User's matric number for identity purpose.	(Matric number - UxxxxxxxA)	UxxxxxxxX. User prompted to enter his email.	PASS	
1-3	(Option 2) - User's email for contact purpose.	(Valid email - test@test.test)	test@test.test. User prompted to enter his course number.	PASS	
1-4	(Option 2) - User enters a valid course number to subsequently choose from the available indexes.	(Valid course number - AAxxxx)	AAxxxx. User prompted to choose his current index.	PASS	
1-5	(Option 2) - User uses the inline keyboard to choose their current index.	Inline keyboard - (index - xxxxx)	Current index: xxxxx. User prompted to choose his wanted index.	PASS	
1-6	(Option 2) - User uses the inline keyboard to choose the index they want.	Inline keyboard - (index - yyyyy)	New index: yyyyy. User prompted with thank you message and has quit from the bot.	PASS	



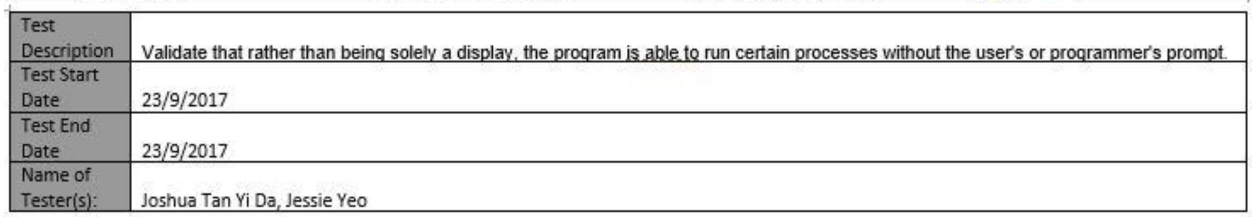
Data validation (incorrect input)					
Test number:	Description:	Input:	Expected output:	Pass/Fail:	Errors/Comments:
2-1	(Option 1) - User enters an invalid course number to view the course indexes.	xxxx	"Invalid course code!", prompted to reenter valid course code.	PASS	
2-2	(Option 1) - User enters an invalid course number to view the course indexes.	AAaaaa	"Invalid course code!", prompted to reenter valid course code.	PASS	
2-3	(Option 1) - User enters an invalid course number to view the course indexes.	(Course number not even in database)	"Invalid course code!", prompted to reenter valid course code.	PASS	
2-4	(Option 2) - User enters an invalid matric number.	UaaaaaaaA	"Invalid matric number!", prompted to reenter valid matric number.	PASS	
2-5	(Option 2) - User enters an invalid matric number.	xxxxxxxA	"Invalid matric number!", prompted to reenter valid matric number.	PASS	
2-6	(Option 2) - User enters an invalid matric number.	UxxxxxxxX	"Invalid matric number!", prompted to reenter valid matric number.	PASS	
2-7	(Option 2) - User enters an invalid matric number.	xxxxxxx	"Invalid matric number!", prompted to reenter valid matric number.	PASS	
2-8	(Option 2) - User enters an invalid matric number.	Uxxxxxxx	"Invalid matric number!", prompted to reenter valid matric number.	PASS	
2-9	(Option 2) - User enters an invalid email.	test@test	"Invalid email!", prompted to reenter valid email.	PASS	
2-10	(Option 2) - User enters an invalid email.	test.test	"Invalid email!", prompted to reenter valid email.	PASS	
2-11	(Option 2) - User enters an invalid email.	test@test(comma)test	"Invalid email!", prompted to reenter valid email.	PASS	
2-12	(Option 2) - User enters an invalid email.	(plain text)	"Invalid email!", prompted to reenter valid email.	PASS	
2-13	(Option 2) - User enters an invalid course number to subsequently choose from the available indexes.	xxxx	"Invalid course code!", prompted to reenter valid course code.	PASS	
2-14	(Option 2) - User enters an invalid course number to subsequently choose from the available indexes.	AAaaaa	"Invalid course code!", prompted to reenter valid course code.	PASS	
2-15	(Option 2) - User enters an invalid course number to subsequently choose from the available indexes.	(Course number not even in database)	"Invalid course code!", prompted to reenter valid course code.	PASS	
2-16	(Option 2) User chooses their current index number.	Typing - xxx (for course index that has 0 at the front)	"Invalid index number!", prompted to choose correct index.	PASS	
2-17	(Option 2) User chooses their current index number.	Typing - index not even in the available options.	"Invalid index number!", prompted to choose correct index.	PASS	
2-18	(Option 2) User chooses their current index number.	Typing - valid index from other courses	"Invalid index number!", prompted to choose correct index.	PASS	
2-19	(Option 2) User chooses the index number that they want	Typing - yyy (for course index that has 0 at the front)	"Invalid index number!", prompted to choose correct index.	PASS	
2-20	(Option 2) User chooses the index number that they want	Typing - index not even in the available options.	"Invalid index number!", prompted to choose correct index.	PASS	
2-21	(Option 2) User chooses the index number that they want	Typing - valid index from other courses	"Invalid index number!", prompted to choose correct index.	PASS	
2-22	(Option 2) User chooses the index number that they want	Typing - xxxxx (current index number)	"Invalid index number!", prompted to choose correct index.	PASS	

Test Description	Verify that the steps in the Telegram bot are run in the right order, as well as to verify that valid but logically incorrect input should lead users to error output.
Test Start Date	23/9/2017
Test End Date	24/9/2017
Name of Tester(s):	Joshua Tan Yi Da

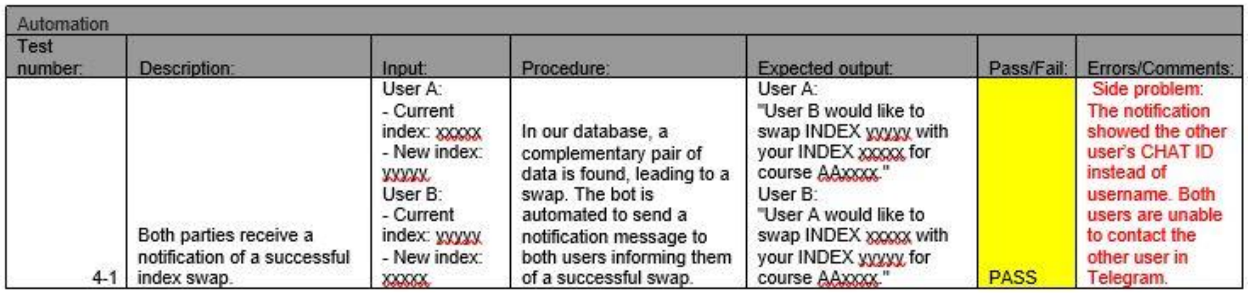
Logic						
Test number:	Description:	Input:	Procedure:	Expected output:	Pass/Fail:	Errors/Comments:
3-1	User is able to start the Telegram bot.	/start	The program is started on the user's Telegram.	(Welcome page + state all features of the Telegram bot)	PASS	
3-2	(Home page) User views the course in interest and all its indexes.	/1	User inputs the first option.	"Please enter your course code"	PASS	
3-3	(Home page) User chooses to swap his current index for another index that he wants	/2	User inputs the second option.	"Please enter your matric number"	PASS	
3-4	(Home page) User views his pending requests for index swapping	/3	User inputs the third option.	(View all the valid pending requests). User subsequently brought to thank you message (to quit).	PASS	
3-5	(Home page) User chooses to cancel one of his pending index swapping request.	/4	User inputs the fourth option.	View all valid pending requests, with the options to cancel a transaction by entering /(associated number)	PASS	
3-6	(Home page) User can quit the bot at the home page.	/quit	User quits the bot at the home page.	(Thank you message, including prompt to /start if users want to restart the bot.)	PASS	
3-7	(Option 2) If user chooses a course with only 1 available index, he has no other options for swapping and should not be able to do so.	Course code: CZ0001	The database should track that those courses do not have more than 1 index available for users to choose their current and then a different index that they want.	Notification saying: "This course only has one index! What are you trying to do?"	FAIL – passed on 24/9/2017	Proceeded to ask user for course index (with no inline keyboard). Upon entering valid course index, bot responded "Wrong input!", and user must force quit.
3-8	(Option 2) After choosing current index, user should not be given the option to choose the exact same index for swapping.	(Chooses current index from inline keyboard)	The database should know that the user would want to choose a different index to swap to, thus not giving him the option to choose his current index again.	When prompted for new index to swap to, option for the current index is not present.	PASS	
3-9	(Option 2) User tries to enter the exact same particulars and requests twice. A duplicate request should not be processed.	(Input exact same particulars twice)	The database should track that users have input the same particulars twice and only one of the requests should be valid.	Go to option 3, only 1 of the request is there.	PASS	
3-10	(Option 2) User enters a second request for the same course that has the combination of (current index: yyyyy and new index: xxxxx). Instead of a swap, this should be an invalid input.	Current index: yyyyy; New index: xxxxx	The database should track that users with the same particulars have entered complimentary sets of data. Logically users wouldn't swap with themselves, thus the 2nd entry should return as an invalid (complimentary) data.	No notification of the swap as no swap should happen. Option 3 only shows one of the requests (which one IDK).	PASS	
3-11	(Option 2) User puts a different current course index for the same course (before any swap happen).	1st current index: xxxxx; 2nd current index: zzzzz	Logically, the user cannot have more than 1 current index at the same time. One of the requests is certainly invalid.	Message to inform user that his 2nd entry is invalid. The 2nd entry is not recorded in the database.	PASS	



3-12	(Option 4) User chooses to cancel an index swapping request. That request should no longer be visible and available.	/4 -> (chooses a number to cancel for that index swapping request).	The database removes the entry for that request, making it unavailable for complementary data to swap with the now cancelled entry.	After cancelling, message to show success in cancelling requests + remaining requests (now excluding the recently cancelled request). User subsequently brought to thank you message (to quit).	PASS	
3-13	(Option 1) Before entering course code, user chooses to quit the bot	Type - /quit	User quits the bot at option 1 - "Please enter your course code."	(Thank you message, including prompt to /start if users want to restart the bot.)	PASS	
3-14	(Option 2) Before entering matric number, user chooses to quit the bot.	Type - /quit	User quits the bot at option 2 - "Please enter your matric number."	(Thank you message, including prompt to /start if users want to restart the bot.). No data recorded.	PASS	
3-15	(Option 2) Before entering email, user chooses to quit the bot	Type - /quit	User quits the bot at option 2 - "Please enter your email."	(Thank you message, including prompt to /start if users want to restart the bot.). No data recorded.	PASS	
3-16	(Option 2) Before entering course code, user chooses to quit the bot	Type - /quit	User quits the bot at option 2 - "Please enter your course code."	(Thank you message, including prompt to /start if users want to restart the bot.). No data recorded.	PASS	
3-17	(Option 2) Before entering current index, user chooses to quit the bot	Type - /quit	User quits the bot at option 2 - "What is your current index?"	(Thank you message, including prompt to /start if users want to restart the bot.). No data recorded.	PASS	
3-18	(Option 2) Before entering new index, user chooses to quit the bot	Type - /quit	User quits the bot at option 2 - "What is your current index?"	(Thank you message, including prompt to /start if users want to restart the bot.). No data recorded.	PASS	
3-19	(Option 4) Before choosing a index swapping request to cancel, user chooses to quit the bot	Type - /quit	User quits the bot at option 4 - (showing all pending request) Quit before cancelling any requests.	(Thank you message, including prompt to /start if users want to restart the bot.) No request for cancelling is recorded.	PASS	



Test Description	Validate that rather than being solely a display, the program is able to run certain processes without the user's or programmer's prompt.
Test Start Date	23/9/2017
Test End Date	23/9/2017
Name of Tester(s):	Joshua Tan Yi Da, Jessie Yeo



Automation						
Test number:	Description:	Input:	Procedure:	Expected output:	Pass/Fail:	Errors/Comments:
4-1	Both parties receive a notification of a successful index swap.	User A: - Current index: xxxxx - New index: yyyyy User B: - Current index: yyyyy - New index: xxxxx	In our database, a complementary pair of data is found, leading to a swap. The bot is automated to send a notification message to both users informing them of a successful swap.	User A: "User B would like to swap INDEX yyyyy with your INDEX xxxxx for course AAxxxx." User B: "User A would like to swap INDEX xxxxx with your INDEX yyyyy for course AAxxxx."	PASS	Side problem: The notification showed the other user's CHAT ID instead of username. Both users are unable to contact the other user in Telegram.

Figures 25 – 33: Results for User Acceptance Testing