The Hong Kong Polytechnic University

Department of Computing

COMP4913 Capstone Project

Report (Final)

An Online Reservation System for Teaching

Student Name:	Yu Fong Yan
Student ID No.:	[redacted]
Programme-Stream Code:	61431-FIS
Supervisor:	Dr Wang QiXin
Co-Examiner:	Dr YANG Yu
2nd Assessor:	Prof. LI Qing
Submission Date:	11 April 2023

Abstract

Being the significant others in offering education to future labor force, teachers are obligated to guarantee their lessons could allow students to acquire certain level of knowledge from their lessons, so they would often assign undergraduates with projects or exams in order to track the learning progresses. Consequently, instructors may need to hold consultation or presentation sessions with students individually or in groups, leading to the concern of timeslots allocation for each party, whose timeslot should be aligned with teachers' schedule without conflicting other peers' appointment. This would require tutor's great effort in compromising with every particular undergraduate, and it would be difficult for them to make the allocation decision in a fast, fair and caring manner without any tool supporting this communication process, especially in the tertiary education context.

Therefore, online reservation system would be essential to resolve this problem, and this report would focus on the purpose, requirements, technologies, implementation of developing this scheduling tool, which aims to be utilized on The Hong Kong Polytechnic University (PolyU).

Table of Content:

Background and Problem Statement

Attending seminars in tertiary institutions, it is observed more than 100 people in the same course is a prevalent practice to be seen. With this great number of students reaching the instructor for one issue individually, the latter would have to tackle with over 100 concerns just for that case, and the same is likely to be put forward with timeslot allocation, and even bring about the fairness problems. For instance, if there is a presentation session among 100 students and no tool would be used in assisting the timeslot assignment, it is necessary for lecturers to communicate with 100 varied students personally for negotiating their preferences manually. Also, they would need to consider what is the time available for both parties to have the meeting? Who would be the first in acquiring the desired scheduled time? Which concerns should be prioritized if there is conflicted selection? This approach would preferably yield heavy time cost, which is less possible for professors to bear it due to their limited spare time and workload.

With the outbreak of COVID-19, the difficulties and cost in implementing the manual method has risen to a drastic degree, particularly in PolyU, which has been adopting hybrid or online teaching mode during the pandemic, meaning that the main communication portals have changed to mails and video conferencing platforms rather than face-to-face. It is found out that this mode may lessen students' engagement [1], resulting in the increment of adversity on connecting with the juniors, as sometimes they would miss messages if they did not pay attention to the computer screen, unlike the confirmation acknowledgement in person during face-to-face lectures.

Indeed, similar products has been appearing in the market, Planyo [2] and Calendy [3] would be the examples, they both could deal with 1:N meeting scheduling problem by offering web

reservation platforms. However, users are only able to choose one timeslot from a list of provided timeslots, meaning that these two systems may not be capable to cope with conflicting timeslot selection in large sample size. Doodle [4], another platform offering users to do polling on different time intervals in knowing people's priorities, but the final allocation results are decided manually, revealing the probability that one could be biased with specific stakeholder and violate the fairness principle.

As a result, automation should be obtained for easing the burden brought by 1:N time allocation decision making, in order to achieve faster, fairer and caring principle when assigning specific timeslot to 100 students by professors in PolyU.

Objectives and Outcome

Addressing the 1:N timeslot allocation problem, this project aims to develop a web meeting reservation system used under PolyU, so as to allow multiple undergraduates to make their bookings with specific professor in an efficient manner, ie. they could all make their preferences at the same time on the web platform, and the result of assigned schedule would be generated automatically after the deadline of selection. In order to fulfil the fairness and caring principle, first-come-first-serve algorithm would be set up for former aspect, and the inputted priorities would be considered beforehand without enforcing mandatory selection and assignment.

Measurable Outcome:

- Improve the existing interactive web application interface provided by Dr Wang
- Update the schedule allocation algorithm to achieve the objectives of efficiency, fairness, and care

- Maintain database in storing all parties' data
- Update database with newest timeslot availability status after each selection round by application itself

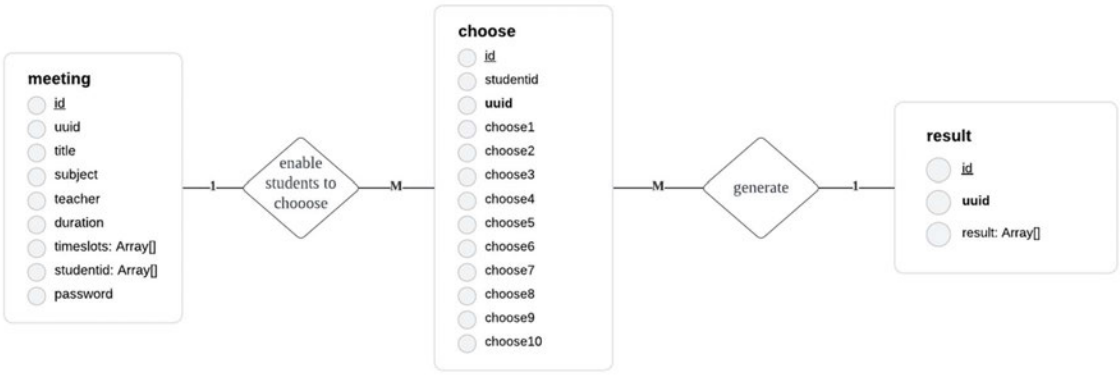
Functional Requirements

- Amendment on existing system
  - Revising the database structure
  - Improve user-friendliness and confidentiality of webpages
  - Update on mapping due to change in algorithm for strengthening fairness
  - First come first serve time allocation
- Automatic declaration of creation, authorization, results through mails sent to students individually
- Generation of the next-round timeslot preference-choosing form
  - if the result could not attain students' preferences in the first round
  - if students are found not filling in the form before the deadline

Project Methodology

1. Database Design

1.1 Previous Work



[Fig.1: database structure from legacy codes]

The database structure from previous work [Fig.1] could show that it could in fact keeping a well 1:M relationships among the tables, which "meeting" table would include all the details needed for creating a meeting with multiple timeslots and multiple students, and each student could choose from the available timeslots in "choose", then multiple allocated timeslots would be assigned to each student after the deadline of selection period in a record from "result" table.

id	uuid	title	subject	teacher	duration	deadline	timeslots	studentid	password
2	58c0dbe2-fac0-49c8-8444-904287191808	1	1	1	10	2023-04-10 00:02:00	["2023-04-09_11:02-11:12","2023-04-09_11:12-11:22"...	["12345678d","12345679d"]	kydz08pu

[Fig.2: "meeting" table structure from legacy codes (example input)]

However, it is noticed there are little limitations from each table design. First, from the "meeting" table [Fig.2], multiple timeslots and studentID (student identification code) tend to be

stored as an array, indeed this could specify teachers' availability and students allowed for a specific exam.

Yet, when it comes to maintainability, data visualization would place a great importance in future development, comparing varied value storing in an array, or storing as individual record in MySQL tables separately, the latter method could be the way to strengthen it.

timeslotid	examid	timeslot	dateid	scheduled
1	0b8247c9-0169-4ab3-a686-313d1dc0247d	2023-04-06_12:10-12:30	1	0
2	0b8247c9-0169-4ab3-a686-313d1dc0247d	2023-04-06_12:30-12:50	1	0

[Fig.3: "meetingtimeslots" table structure from current work (example input)]

If one is doing testing whether specific timeslot in storing in the database, functions could be used for scanning the array for retrieving the answer, but if the timeslot is individual record, sometimes commands are not needed because of the organized separation of timeslot records are clearly shown on the table [Fig.3], resulting in a faster acknowledge of testing result, and maintainability could be increased.

id	uuid	result
1	58c0dbe2-fac0-49c8-8444-90428719180	{"2023-04-09_11:02-11:12":"12345678d","2023-04-09_11:12-11:22":"12345679d","2023-04-09_11:22-11:32":0,"2023-04-09_11:32-11:42":0,"2023-04-09_11:42-11:52":0,"2023-04-09_11:52-12:02":0}

[Fig.4: "result" table structure from legacy codes (example input)]

Second, "result" table also adopted array in storing assignment results, and it acts as dictionary in mapping student with timeslot [Fig.4], which could be extended to three problems. One would be data readability that owns the similar concerns as above, another would be little memory wastage, perhaps be attributed to the involvement of non-allocated timeslot in this array, which "meeting" has already stored all of them. In addition, data security could be harmed during the

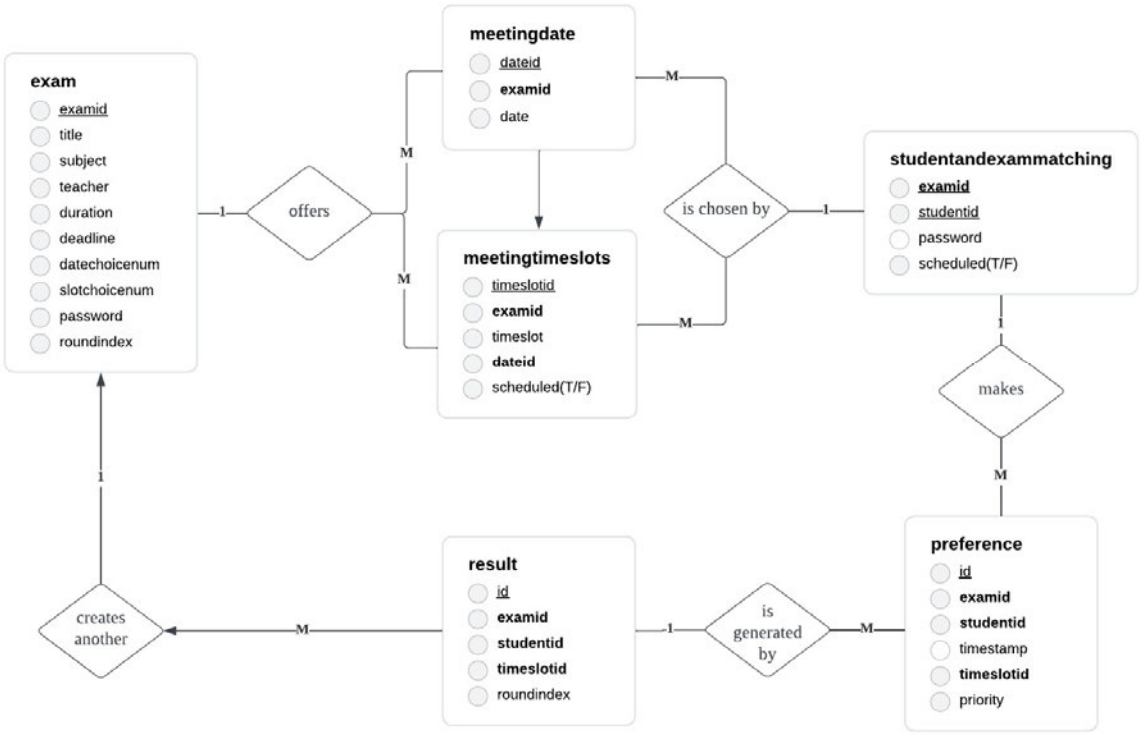
second round of timeslot result generation, as it is necessary to take out the whole array for updating students' mapping, indicating that the already-settled timeslots and studentID would be pulled out for operation again, this would add the risk of data loss with program accessibility in running through all the data in an array.

id	studentid	uuid	choose1	choose2	choose3	choose4	choose5	choose6	choose7	choose8	choose9	choose10
1	12345678d	58c0dbe2-fac0-49c8-8444-904287191808	2023-04-09_11:02-11:12	2023-04-09_11:12-11:22	2023-04-09_11:12-11:22	2023-04-09_11:02-11:12	2023-04-09_11:02-11:12	2023-04-09_11:12-11:22	NULL	NULL	NULL	NULL

[Fig.5: "choose" table structure from legacy codes (example input)]

Third, from the "choose" table [Fig.5], the inflexibility of database structure could be viewed with the NULL values on the columns of putting in the priority of timeslot choices, the previous work has fixed students with only at most 10 options as their preferences input, which has limited the scalability of program, owing to teachers may want to enable students to add more than 10 preferences for his references. While storing NULL values could be a dangerous move with lowering down the reliability of system, as it is possible that null values could terminate the system operation with some specific functions that eliminate the use of null, also memory is wasted in this case.

1.2 New approach



[Fig.6: current database structure]

Therefore, new approach of database structure [Fig.6] is suggested in this study to lessen the mentioned reflection. This design would still maintain a 1:M relationship with the connection of tables, “exam” table would still store all the meeting details needed, and multiple timeslots would be stored in “meetingtimeslots” table, individual timeslot would be identified with its timeslotid, as one would choose the preferred date before choosing preferred timeslots, so “meetingdate” table would be storing available date preferences, both tables would map with corresponding examid (meeting session identification code). Then, multiple students would be stored in “studentexammatch” table, which individual student would be identified with own studentid, and mapped with corresponding examid.

id	examid	studentid	timestamp	timeslotid	priority
13	cd94a10e-daff-4b35-b458-382ba7e879d4	clarissa112y	1680795547	16	1
14	cd94a10e-daff-4b35-b458-382ba7e879d4	clarissa112y	1680795547	17	2

[Fig.7: “preference” table structure from current work (example input)]

Then, the per-date timeslot selection preferences input would be stored in “preference” table for each student and each priority, ie. format indication in Fig.7. Result will be thus generated based on the students’ preferences, and each record will represent only one student’s specific scheduled timeslot. After knowing the allocation in the first-round selection, some may still have not yet assigned with specific time, then another round of selection would required to be created, so exam table would have roundindex value as the representation of number of rounds for the current timeslot preference selection process.

How could this structure eliminate the issues stated?

examid	studentid	password	scheduled
5619a474-338b-4343-936f-90c4ad0085d9	19065597D	dm93pacp	0
5619a474-338b-4343-936f-90c4ad0085d9	cla112y	m3ipoksq	0

[Fig.8: “studentexammatch” table structure from current work (example input)]

id	examid	studentid	timeslotid	roundindex
1	cd94a10e-daff-4b35-b458-382ba7e879d4	19065597D	9	1
2	cd94a10e-daff-4b35-b458-382ba7e879d4	cla112y	10	1

[Fig.9: “result” table structure from current work (example input)]

First, concerning the data visualization, it is easier to read the data by separating the array values into record, changes could be viewed in Fig.8. Second, scheduled attribute in “preference” and “studentexammatch” [Fig.3, Fig.8] could store 0 as unscheduled and 1 as scheduled that implies

whether the student has been assigned with specific timeslot, and timeslot has been utilized or not. Then, new “result” table [Fig.9] just needed to store the scheduled result instead of unscheduled, and the need in pulling out whole array for editing the result would be reduced.

examid	title	subject	teacher	duration	deadline	datechoicenum	slotchoicenum	password	roundindex
0b8247c9-0169-4ab3-a686-313d1dc0247d	realtest	realtest	realtest	20	2023-04-07 00:10:00	2	3	j80o5utk	1

[Fig.10: “exam” table structure from current work (example input)]

Third, addressing the inflexibility on “choose” [Fig.5], “exam” [Fig.10] table will store datechoicenum and slotchoicenum to allow teachers to input how many preferred days and preferred timeslots for each day available for students’ input options, such as 2 and 3 respectively, then there would be total of 2 date preferences and 2*3 = 6 timeslot choices for student option form, and 6 preferred timeslots would be stored in “preference” table [Fig.7] with priority stated, which could refrain from applying NULL values and the fixed option number of 10 in “choose”.

2. User Flow Design

Preparation Stage: Request from Teacher to Application

- Put forward the time available for meeting
- Specify the meeting details, including meeting length, students needed and deadline for the registration
- Divide the schedule into timeslots automatically by web application
- Generate url or identification code for students' registration in particular course meeting by online method

Stage 1: Request from Student to Teacher

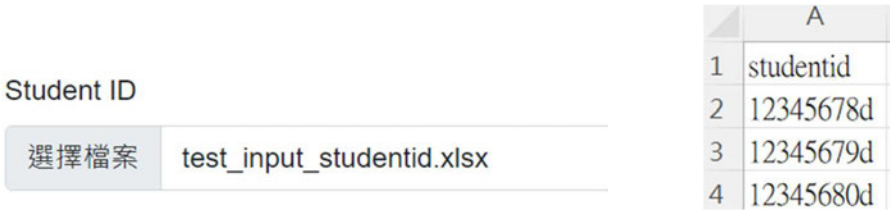
- Click the url distributed or enter the identification code on the web system for registration
- Choose the top n number of desired time preference
- Submit the online form to teacher for further arrangement
- Edit the preference list if essential

* The above 2 stages have been completed from the legacy codes provided from Dr Wang.

Amendments and Added features (New approach)

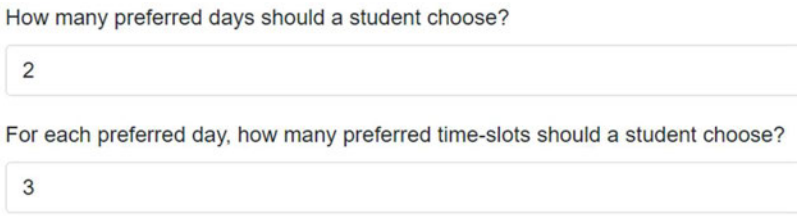
The following parts would reveal the altered and additional function and codes from the current developed version in this project.

Preparation Stage: Request from Teacher to Application



[Fig.10: input field of studentid from creatingmeeting.php (example input)]

- During the preparation of creating exam session form, if professor wants to input the studentid, he no longer requires typing the codes on text field and separate with commas but inputting the excel file with formats shown in Fig.10, PhpSpreadsheet framework would be taking care of reading and writing it into database. This change would increase users' usability with higher user-friendliness.

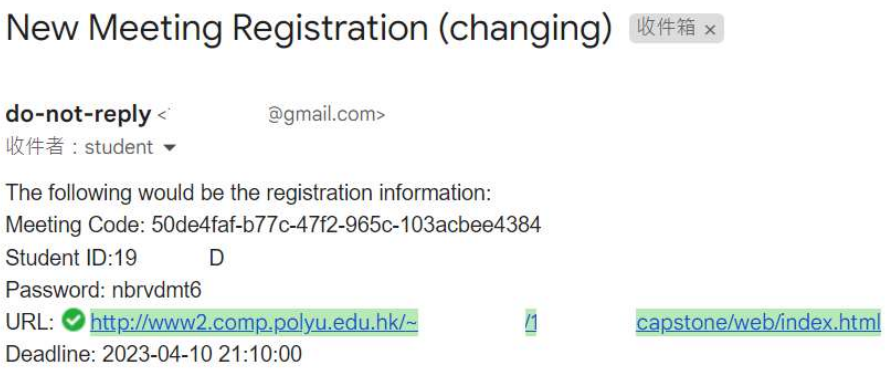


[Fig.11: input field of number of date and timeslot preferences (example input)]

- Moreover, the number of "n" date preferences, and number of "m" timeslots available per date would be defined in the form [Fig.11]. Timeslots would not be the only factor of consideration on choosing desired exam time, *per-date timeslot preference* would be concerned instead, implying that number of *(i) preferred days* and *(ii)timeslot for that day* would have to be noticed for student preference input. The reason why this would be implemented is that sometimes students are inclined to choose the last date with latest timeslot for exams, multiple conflicts among students' timeslots would then exist. Thus, *preferred days* should be selected first, more details would be shown on Stage 1.

[Fig.12: send mail button on status.php (example input)]

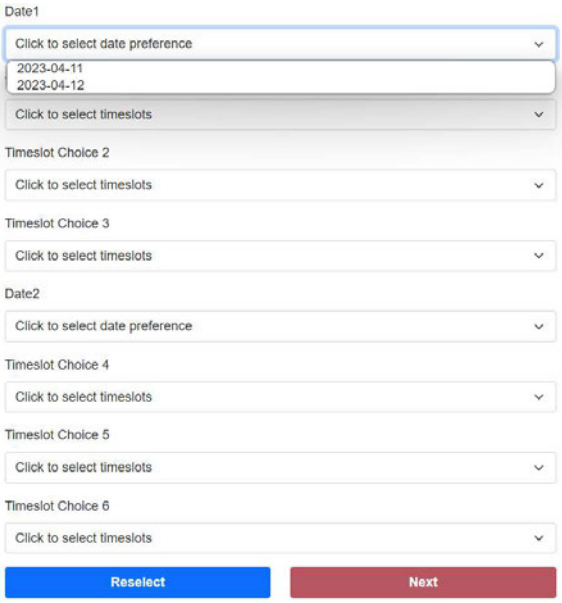
- After teacher submitted the form, meeting code, teacher's edit password, details would be shown on a new page, complementing with this new button [Fig.12], this "Send Mail" button could be clicked to automatically sending out emails to students with meeting code and their personal password for authorizing in the time selection.



[Fig.13: mail received by individual student (example input)]

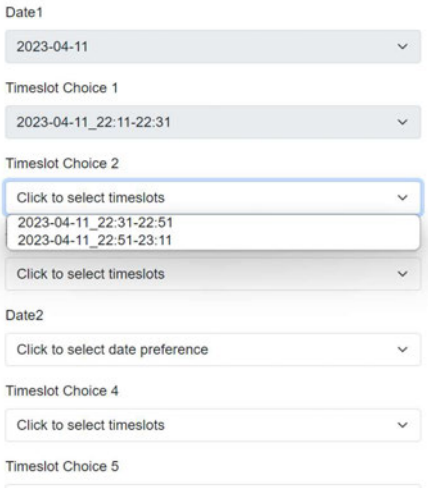
- When students obtain these details, they could click into the URL for inputting their preferences. Password authorization would be added for student log in before their selection, as students could fake other's identity in choosing their timeslot without it, which may cause unfairness, and PhpMailer framework would be supporting this function.

Stage 1: Request from Student to Teacher



[Fig.14: timeslot selection form (example input)]

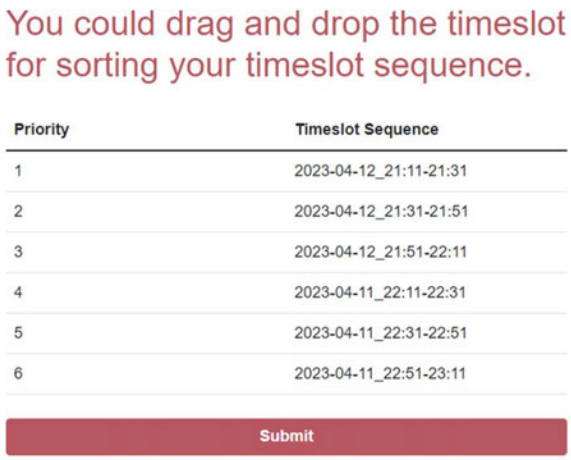
- When student entered meeting code, studentid and own password, selection form would be displayed [Fig.14], and it provide "n" *preferred days* and "m" *timeslot per date* as stated in preparation stage. If number of *preferred days* (n) is 2 and *timeslot per date* (m) is 3 [Fig.11], Fig.14 would be the interpretation for the preference form based on these two values, ie. Date1,2 and Timeslot1,2,3 for Date 1, Timeslot4,5,6 for Date 2.



[Fig.15: timeslot selection form (example input)]

- Undergraduate could only first choose one preferred date, then timeslot options would be available for them to choose their preferred timeslot for that date [Fig.15], ajax codes would

be utilized to complete this function in receiving students' response on the date preference first.



[Fig.16: timeslot selection form (example input)]

- If they completed the form and click "Next" button [Fig.14], they still have to sort the sequence of the timeslot they selected [Fig.16]. Imagining this would be a larger sample size, 30 timeslots availability has been given by a student in a mandatory manner, it is possible that one could have few unavailable timeslots for specific date, could be 2-3 in this sample size, then this page could allow that student to drag and drop the timeslot to lower position for her lower priority for that specific timeslot, rather than forcing one to have higher priority just because of the chosen preferred date, fulfilling the caring objective. Preference priority would be updated in the database once "Submit" button is clicked, the timestamp of students completing the form would also be stored and updated in "preference" table for mapping in the next stage.

Stage 2: Request from Teacher to Student

- Mapping students to scheduled time with first-come-first-serve principle after deadline
- Generate allocation results automatically

```
$studentidarray_random= $studentidarray;
shuffle($studentidarray_random);
```

[Fig.16: allocation algorithm from previous work]

- Owing to the shuffling of students' sequence no matter who insert the record first in previous work [Fig.16], "first-come-first-serve" is stressed in the current work to preserve the fairness.

* Therefore, the main tasks for Stage 2 would be updating of allocation algorithm along with codes that contributed by the previous work.

```
$stmt = $conn->prepare("SELECT * FROM `preference` WHERE `examid` = ? GROUP BY `studentid` ORDER BY `timestamp` ASC;");

$stmt->bind_param("s" , $_GET['examid'] );

$stmt->execute();

$result = $stmt->get_result();
```

[Fig.16: allocation algorithm from current work]

- The new practise could be viewed on the SQL command written on Fig.16, which records of preferences would be order by the timestamp, ie. time that specific student has handed in the form, in ascending order, as larger value of unix timestamp would be implying the time later. Sequence of student would then be sorted by their form input time in order to achieve first-come-first-serve principle.

```
$notscheduled0 = 0;
$stu = $conn->prepare("SELECT * FROM `studentexammatch` WHERE `examid` = ? AND `studentid` = ? AND `scheduled` = ?");
$stu->bind_param("ssi" , $_GET['examid'],$row['studentid'], $notscheduled0);
$stu->execute();
$sturesult = $stu->get_result();
```

[Fig.17: allocation algorithm from current work]

```
$notscheduled = 0;
$timeselection = $conn->prepare('SELECT * FROM `meetingtimeslots` WHERE `examid` = ? AND `scheduled` = ?');
$timeselection->bind_param("si", $_GET['examid'], $notscheduled);
$timeselection->execute();
$selectedR = $timeselection->get_result();
```

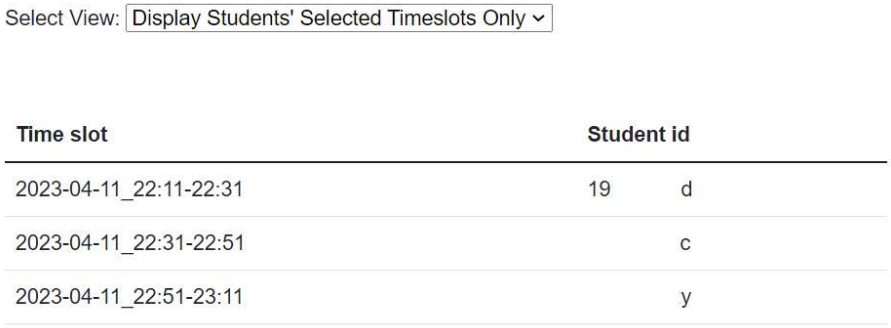
[Fig.18: allocation algorithm from current work]

```php
$num = 1;
$scheduled = $conn->prepare('UPDATE `studentexammatch` SET `scheduled` = ? WHERE `examid` = ? AND `studentid` = ?;');
$scheduled->bind_param("iss", $num, $_GET['examid'] ,$value);
$scheduled->execute();

$scheduledT = $conn->prepare('UPDATE `meetingtimeslots` SET `scheduled` = ? WHERE `timeslotid` = ?;');
$scheduledT->bind_param("is", $num ,$slotid);
$scheduledT->execute();
```

[Fig.19: update “scheduled” index status]

- After that, only not scheduled students and timeslots would be pointed out for the specific exam meeting timeslot allocation to do the mapping by searching record with “0” in scheduled index, once they are being mapped, their scheduled status would be updated as occupied with “1” [Fig.19].



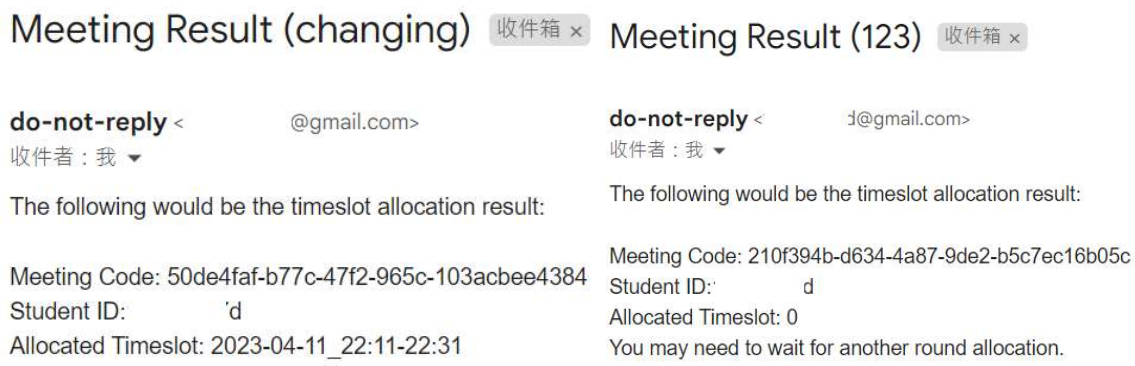
[Fig.20: “display students’ mapped preference” view in showing results to teacher]

[Fig.21: “display all timeslots” view in showing results to teacher]

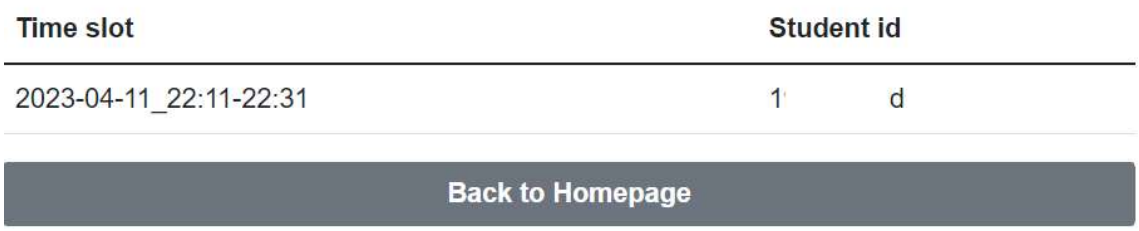
- Following by the update, allocation mapping of one student acquiring one timeslot would be inserted into the database and show in the web platform [Fig.20, Fig.21] with 2 display views, and this result-checking page would include “Send Mail” button for automate result declaration to specific students.

Stage 3: Request from Teacher to Student

- Declare results to students about the arrangement through mail
- Generate forms and its url for second round allocation with updates on timeslots availability
- Send another mail to students who did not obtain certain assignment of timeslots to go through Stage 1 to Stage 3 again until all could obtain successful scheduling result

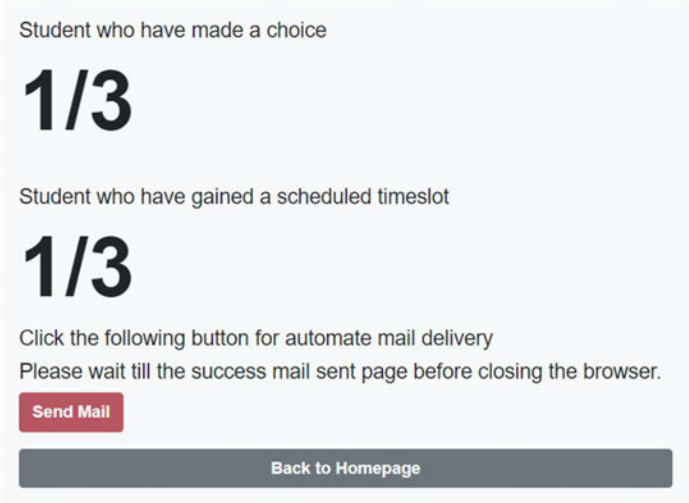


[Fig.22: mail received by individual student (example input)]



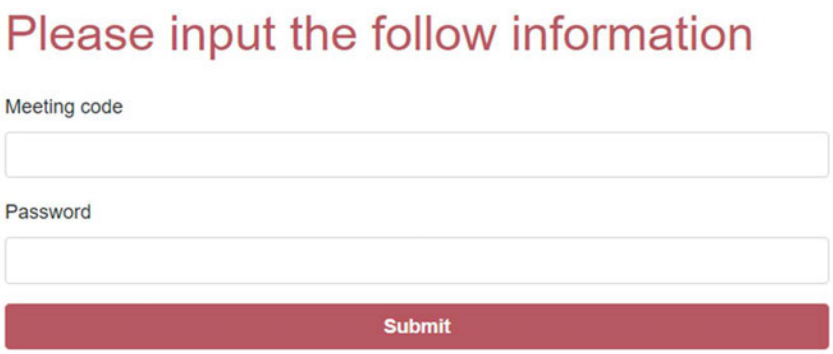
[Fig.23: mail received by individual student (example input)]

- Students would receive the mail no matter they would be allocated with particular timeslot, they could visit the web platform again, and the page would only show his result for own reference [Fig.23], as studentid and password are added to require one in checking result.



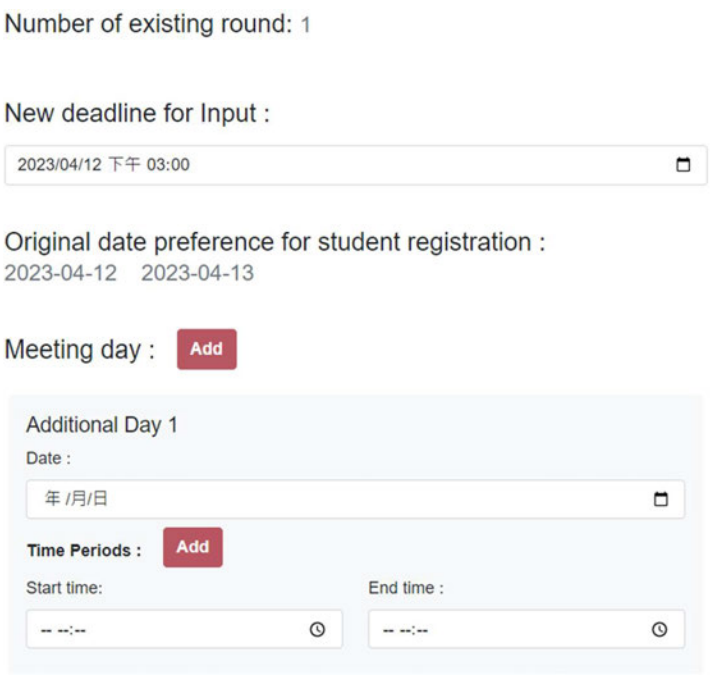
[Fig.24: check status page]

- Status page [Fig.24] could allow teacher acknowledging whether there are students have not yet gained a scheduled timeslot, if that is the case, then the generation of another round of exam timeslot selection would be demanded.



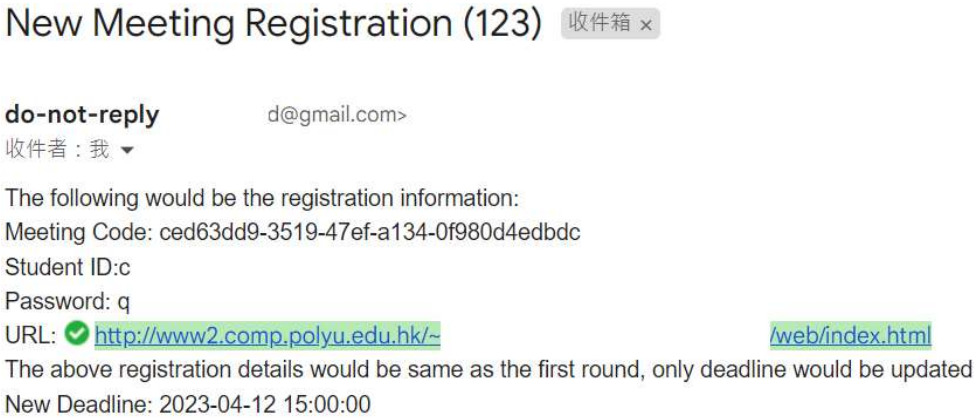
[Fig.25: creation of another round of preference selection (example input)]

- Teachers are required to input the meeting code and password of existing exam scheduling [Fig.25], in order to gain access to the creation form for another round of student choices selection, as this registration details should only be kept by teacher, and the design that creating another round would not generate a new meeting code, but only updating the values in the original database, ie. details in “exam” table, such as roundindex, new deadline for other rounds [Fig.26].



[Fig.26: creation of another round of preference selection (example input)]

- Besides, meeting day and timeslot [Fig.26] could be added optionally if instructor wanted to provide more availability for students' choice, and this would be inserting records into the "meetingdate" and "meetingtimeslots" table.



[Fig.27: creation of another round of preference selection (example input)]

- Mail could be automatically sent when instructor clicked "Send Mail" button again, this could show that students' registration details and examid would be remain unchanged for usability and maintainability, as changing information each time for one registered would be

probably confusing in student's perception, they would need to ensure they are still selecting timeslots from that specific exam, which examid is the representation, so it should be kept being the same. Also, how could the remaining timeslots be tracked by the new examid generated? It is less possible for the system to notice the connection of first and other rounds of allocation for specific exam.

- After student receiving the details again, the next round of choices input will start again, so stage 1-3 will be repeated until teachers have assigned all the students with own allocated result.

3. Deployment

Owing to the utilization under PolyU context, the web should be guaranteed to be run in PolyU intranet, so codes would be updated in the webhome directory in PolyU server, and the web platform would be available at the URL:

http://www2.comp.polyu.edu.hk/~[redacted]/[redacted]capstone/web/index.html

Evaluation

To understand how the system improved after the amendment and development this year, few evaluation metrics would be essential to study its effectiveness.

Requirements fulfillments

"Automation" is the key that this online reservation system was made, so it is essential to fulfill the functional requirements that set in this project, automate creation of available exam sessions, automate generation of authorization details and mapping results, automate mail sent for exam and result details declaration.

1. **Automate creation of available exam sessions**

   With the legacy code provided, exam sessions could be generated for students' registration, and complementing with the current code's amendment, more rounds could be created to allow remaining timeslots to be assigned with unscheduled students, which could maintain a more carer allocation system.

2. **Automate generation of authorization details and mapping results**

   Exam authorization details could be automatically generated from the previous work, and the current approach would be adding student's authorization details for the concern of information security, which would be mentioned in following session. While result mapping could be done in randomized sequence of students in previous work, this part would be replaced by the updates in first-come-first-serve timeslot allocation algorithm for fairness.

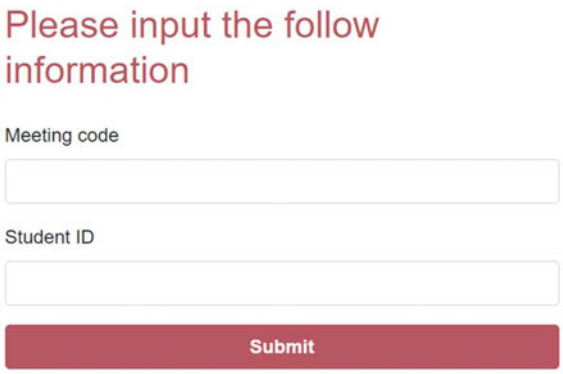
3. **Automate mail sent for exam and result details declaration**

   Previous work did not cover this part, this new approach has covered individual mails sent to particular student about his exam session and authorization details, or mapped result after result is generated. However, this project has not yet cover automatic mail sent to teacher,

although they just need to mark down the exam code and edit password, this would be feasible practise to remind them the authorization details and passed deadline of timeslot allocation for them to check the result.

Information security

In terms of service information security, refer to Stallings [5], a service could yield security problems on “confidentiality”, “integrity”, “availability” and “authenticity”, when one is sending information from source to destination.



[Fig.28: log in page for choice input, choice check and result page from legacy code (example input)]

It is mentioned that “threat” exists when there is another user who knows the information sent during the data transmission at “confidentiality” aspect [5]. On the legacy work, it is noticed that students could gain access to few functions just by entering the meeting code and studentid, which could be easily obtained by checking the campus mailbox, revealing that it is a highly exposed code that could be obtained by great number of students and teachers.

This would increase the exposure of individual student’s private information to other peers to a large extent, especially on their own preference choices and mapped result [Fig.28]. The reason is that the above-mentioned log-in requirements would be adopted in (i) checking student’s personal choices (ii) checking results and (iii) preference input form pages.

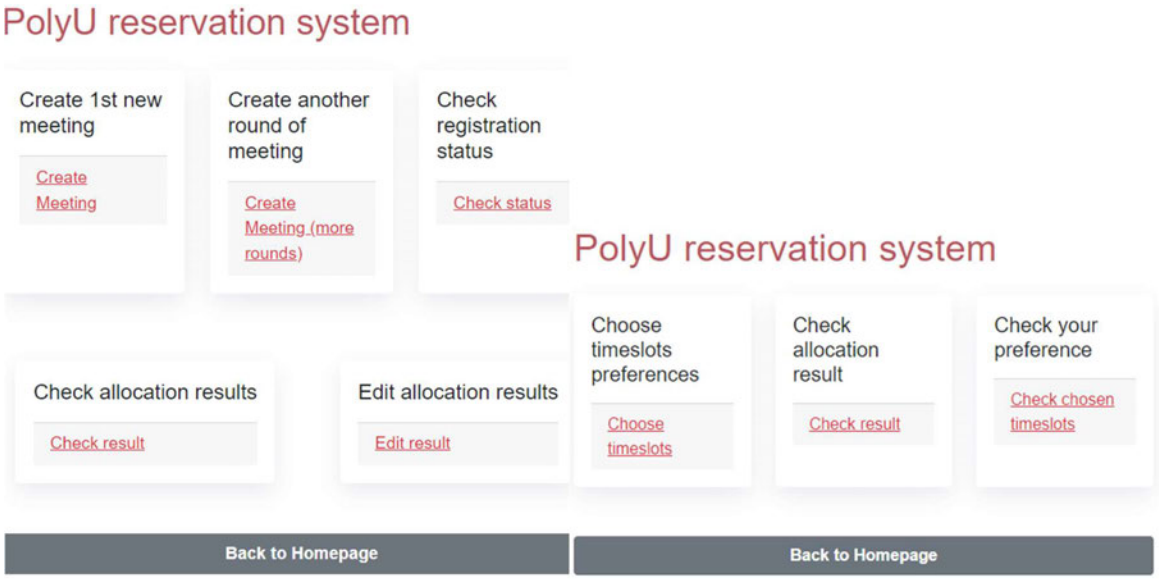
“Confidentiality” is harmed by user’s ability in reading other’s timeslot selection and mapped results, as one has the right to not share these details to others, and “integrity” and “authenticity” is also deteriorated with user’s capabilities in “modifying” and “writing” [5] preferences in the timeslot priority selection forms respectively. These two fields are highly concerned, as they are related to the vital objective that holds in this project, ie. fairness.

The fact that other people could modify and write own preference input is unfair and unethical move to every student, but the lack of password has increased this risk and its occurrence, as faking identity could be done easily without acknowledgement in this log-in manner, particularly user could alter the choices unlimitedly before the deadline, it is probable that student could obtain undesired timeslot with this situation.

Therefore, (i) checking student’s personal choices, (ii) checking results, and (iii) preference input form pages, three of them has already been altered with authentication in the current work. Pages about preferences checking and choosing, student password would be added for checking identities, as mentioned above, mails would be sent to inform student personally, such to uphold “integrity” and “authenticity” purpose. While result-checking page [Fig.20, Fig.21, Fig.23] would require teacher or student password for allowing the former the class allocation result and the latter own timeslot allocation result for upholding “confidentiality” purpose and protect information security in current web service.

Usability

[Fig.29: index.php]



[Fig.30: teacher's and students' view respecitively]

When one enters the homepage, they would click either "teacher" or "student" button shown on Fig.29, and it would lead to the display of Fig.30, which separated views would be given to teacher and student to address what functions could be used with their own identity. Few "back to homepage" buttons would be added for user-friendliness. While user flow design has also introduced few amended features, such as teacher's input on multiple studentid, or two result mapping views for teachers, would increase user's reduced effort in inputting and view the information needed.

Reliability

Data validation has not yet fully accomplished in the work, as it serves as the second priority behind the functions, so it is needed for strengthening them in future work.

Resources Utilization

Hardware

- Computer for web development
- Web server (provided by PolyU)

Software

- PhpMyAdmin
- MySQL
- Php
- Apache
- WinSCP (accessing personal page in PolyU server)

References

[1] B. Hollister, P. Nair, S. Hill-Lindsay, and L. Chukoskie, “Engagement in Online Learning: Student Attitudes and Behavior During COVID-19,” Frontiers in education (Lausanne), vol. 7, 2022, doi: 10.3389/feduc.2022.851019.

[2] Planyo. “THE MOST FLEXIBLE RESERVATION SYSTEM YOU'LL EVER FIND.” Online reservation system & booking software Planyo. https://www.planyo.com/ (accessed Oct.20, 2022).

[3] Calendy. “Easy scheduling ahead.” Calendly: Free Online Appointment Scheduling Software. https://calendly.com/ (accessed Oct.20, 2022).

[4] Doodle. “Professional scheduling made easy.” Free online meeting scheduling tool | Doodle. https://doodle.com/en/ (accessed Oct.20, 2022).

[5] W. Stallings. “Security Services” in Cryptography and Network Security, 7th ed. Pearson Education Limited, 2017, ch.1 pp. 29- 32