COURSE ARRANGER

Project Code: H06

Group members:

DO Van Quyet_20561634_vqdo@connect.ust.hk (Leader)

DO Thuy Trang_20549272_ttdo@connect.ust.hk

PHAM Trung Kien_20553388_tkpham@connect.ust.hk

COMP 2012H

Dr. TSOI Yau Chat Desmond

Table of content

Project Objectives

Inspired by UST Space, Course Arranger is a user-based project that has been designed and developed to help HKUST students efficiently schedule their weekly timetable of the upcoming semester, dealing with several time-consuming problems which often happen during the add-drop period such as class sessions overlapping or credit overloading… Based on the students' inputs, Course Arranger will be capable of:

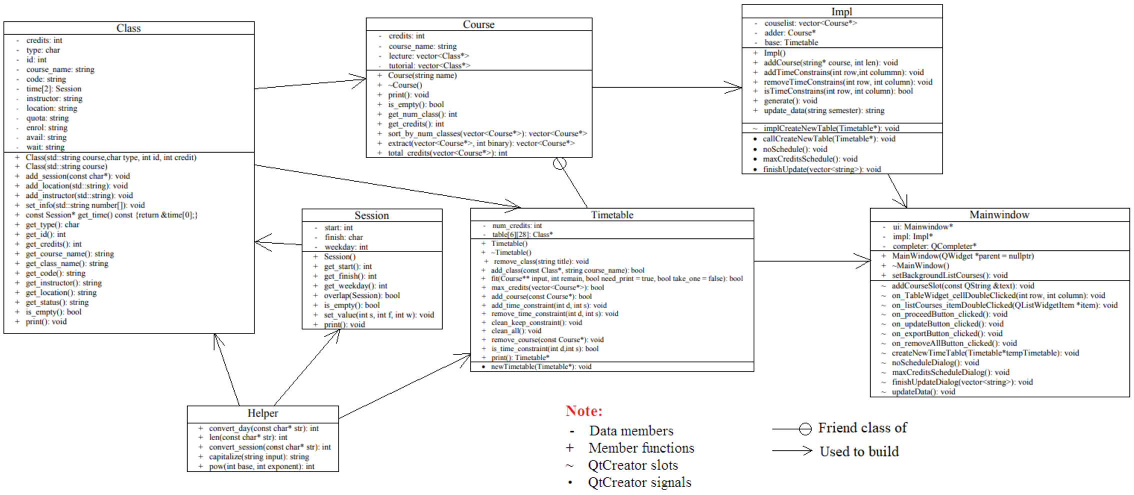
1. Generating all possible timetables that consist of all desired courses within given time constraints, which every HKUST students can gain benefits from.
2. If there always exist sessions clashing between different courses or with time constraints, suggesting a timetable with the highest possible number of credits that could be especially useful for those who only need credits to graduate or fulfill school requirements.

External library

We use a header file **urlmon.h** and the corresponding library file **liburlmon.a** to use **URLDownloadToFile()** as the tool to download data. They are parts of **Microsoft Windows SDK** (Software Development Kits).

OOP Design

Inspect the image OOP_Design.png to see the full structure. Remark that Helper is a group of functions helping the implementation of other built-in classes.

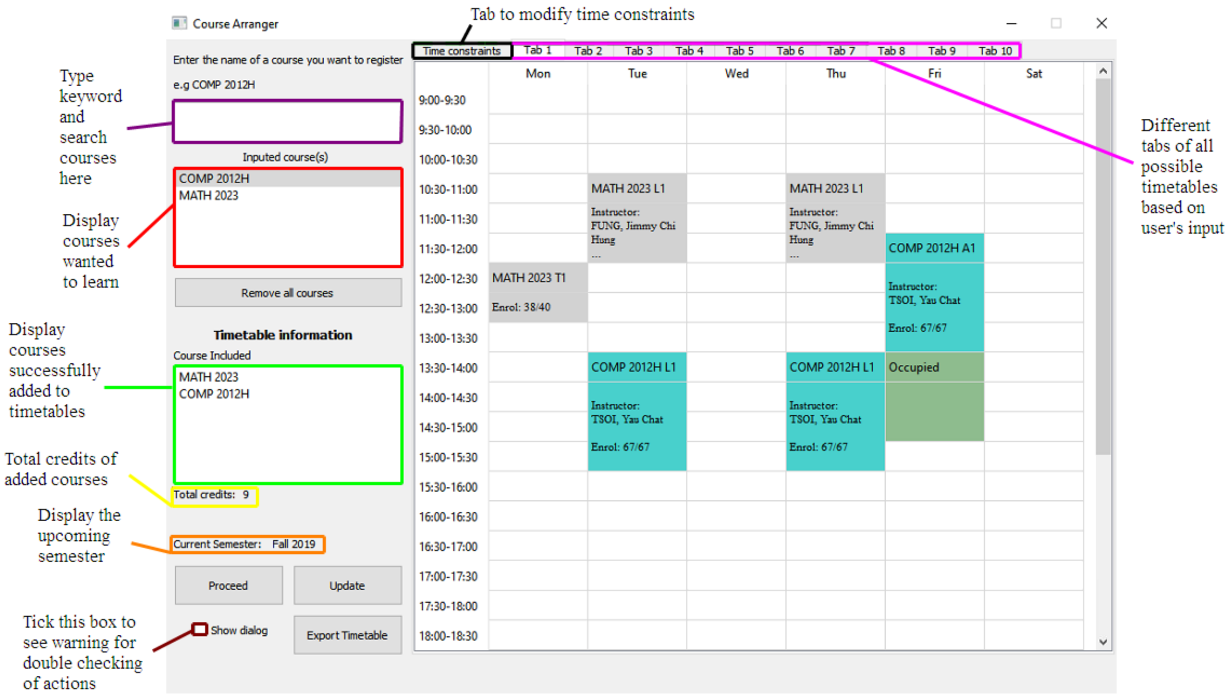

Class
- credits: int
- type: char
- id: int
- course_name: string
- code: string
- time[2]: Session
- instructor: string
- location: string
- quota: string
- enrol: string
- avail: string
- wait: string
+ Class(std::string course,char type, int id, int credit)
+ Class(std::string course)
+ add_session(const char*): void
+ add_location(std::string): void
+ add_instructor(std::string): void
+ set_info(std::string number[]): void
+ const Session* get_time() const {return &time[0];}
+ get_type(): char
+ get_id(): int
+ get_credits(): int
+ get_course_name(): string
+ get_class_name(): string
+ get_code(): string
+ get_instructor(): string
+ get_location(): string
+ get_status(): string
+ is_empty(): bool
+ print(): void
Course
- credits: int
- course_name: string
- lecture: vector<Class*>
- tutorial: vector<Class*>
+ Course(string name)
+ ~Course()
+ print(): void
+ is_empty(): bool
+ get_num_class(): int
+ get_credits(): int
+ sort_by_num_classes(vector<Course*>): vector<Course*>
+ extract(vector<Course*>, int binary): vector<Course*>
+ total_credits(vector<Course*>): int
Impl
- couselist: vector<Course*>
- adder: Course*
- base: Timetable
+ Impl()
+ addCourse(string* course, int len): void
+ addTimeConstrains(int row,int colummn): void
+ removeTimeConstrains(int row, int column): void
+ isTimeConstrains(int row, int column): bool
+ generate(): void
+ update_data(string semester): string
~ implCreateNewTable(Timetable*): void
• callCreateNewTable(Timetable*): void
• noSchedule(): void
• maxCreditsSchedule(): void
• finishUpdate(vector<string>): void
Session
- start: int
- finish: char
- weekday: int
+ Session()
+ get_start(): int
+ get_finish(): int
+ get_weekday(): int
+ overlap(Session): bool
+ is_empty(): bool
+ set_value(int s, int f, int w): void
+ print(): void
Timetable
- num_credits: int
- table[6][28]: Class*
+ Timetable()
+ ~Timetable()
+ remove_class(string title): void
+ add_class(const Class*, string course_name): bool
+ fit(Course** input, int remain, bool need_print = true, bool take_one = false): bool
+ max_credits(vector<Course*>): bool
+ add_course(const Course*): bool
+ add_time_constraint(int d, int s): void
+ remove_time_constraint(int d, int s): void
+ clean_keep_constraint(): void
+ clean_all(): void
+ remove_course(const Course*): void
+ is_time_constraint(int d,int s): bool
+ print(): Timetable*
• newTimetable(Timetable*): void
Mainwindow
- ui: Mainwindow*
- impl: Impl*
- completer: QCompleter*
+ MainWindow(QWidget *parent = nullptr)
+ ~MainWindow()
+ setBackgroundListCourses(): void
~ addCourseSlot(const QString &text): void
~ on_TableWidget_cellDoubleClicked(int row, int column): void
~ on_listCourses_itemDoubleClicked(QListWidgetItem *item): void
~ on_proceedButton_clicked(): void
~ on_updateButton_clicked(): void
~ on_exportButton_clicked(): void
~ on_removeAllButton_clicked(): void
~ createNewTimeTable(Timetable*tempTimetable): void
~ noScheduleDialog(): void
~ maxCreditsScheduleDialog(): void
~ finishUpdateDialog(vector<string>): void
~ updateData(): void
Helper
+ convert_day(const char* str): int
+ len(const char* str): int
+ convert_session(const char* str): int
+ capitalize(string input): string
+ pow(int base, int exponent): int
Note:
- Data members
+ Member functions
~ QtCreator slots
• QtCreator signals
Friend class of
Used to build

Data structures used

- Vector of Pointers: used to store “Class” and “Course” pointers for some purposes with several demands that Vector of Pointer satisfies.
    - Resizable: The data is not uniform and bounded. The number of sections in a course varies from 1 to more than 20, the number of courses given by users is unpredictable.
    - Sortable: Our algorithm is attempting to add each course to the base table respectively. If we sort courses in ascending order of the number of sections, the total number of cases we need to check **until the first time all courses can be added to the base table** is **smaller**. That number is **constant** if we want **all valid (no time crashing) schedules** consisting of all given courses (of course time constraints are considered)
    - Middle-part accessible: When a set of courses is **not fittable**, i.e no valid schedule, we need to **extract subsets of courses** to manipulate and return the timetable with the highest possible number of credits.
    - Fast manipulation: **Objects** of classes “Course” and “Class” are **big**, so pointers play as the light **representatives**.
- Array (1D and 2D): Using 1D arrays for sessions inside “Class” because each “Class” seems to have at most 2 sessions per week (**uniform**). Besides, we use a 2D array of “Class” pointers inside “Timetable” as the **simulation** of the real timetable.

Features incorporated

Course Arranger's Interface:



Tab to modify time constraints

Type keyword and search courses here

Display courses wanted to learn

Display courses successfully added to timetables

Total credits of added courses

Display the upcoming semester

Tick this box to see warning for double checking of actions

Different tabs of all possible timetables based on user's input

Enter the name of a course you want to register

e.g COMP 2012H

Inputed course(s)

COMP 2012H
MATH 2023

Timetable information

Course Included

MATH 2023
COMP 2012H

Total credits: 9

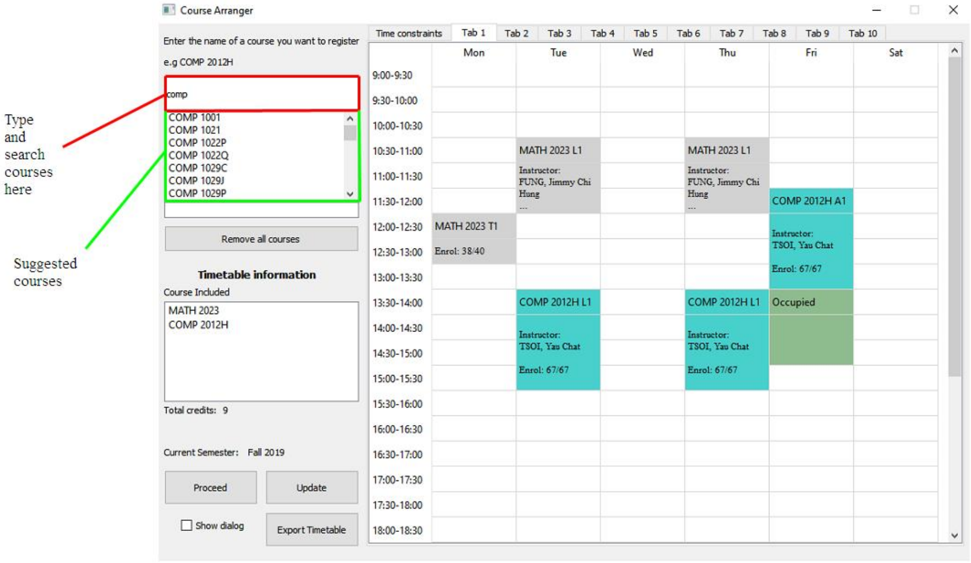
Current Semester: Fall 2019

	Mon	Tue	Wed	Thu	Fri	Sat
9:00-9:30						
9:30-10:00						
10:00-10:30						
10:30-11:00		MATH 2023 L1		MATH 2023 L1		
11:00-11:30		Instructor: FUNG, Jimmy Chi Hung		Instructor: FUNG, Jimmy Chi Hung		
11:30-12:00		...		...	COMP 2012H A1	
12:00-12:30	MATH 2023 T1				Instructor: TSOI, Yau Chat	
12:30-13:00	Enrol: 38/40				Enrol: 67/67	
13:00-13:30						
13:30-14:00		COMP 2012H L1		COMP 2012H L1	Occupied	
14:00-14:30		Instructor: TSOI, Yau Chat		Instructor: TSOI, Yau Chat		
14:30-15:00		Enrol: 67/67		Enrol: 67/67		
15:00-15:30						
15:30-16:00						
16:00-16:30						
16:30-17:00						
17:00-17:30						
17:30-18:00						
18:00-18:30						

Key features:

1. Add courses:

Type keywords of courses (e.g. COMP, MATH) to search and subsequently add to a list.



Type and search courses here

Suggested courses

Enter the name of a course you want to register

e.g COMP 2012H

comp

COMP 1001
COMP 1021
COMP 1022P
COMP 1022Q
COMP 1029C
COMP 1029J
COMP 1029P

Timetable information

Course Included

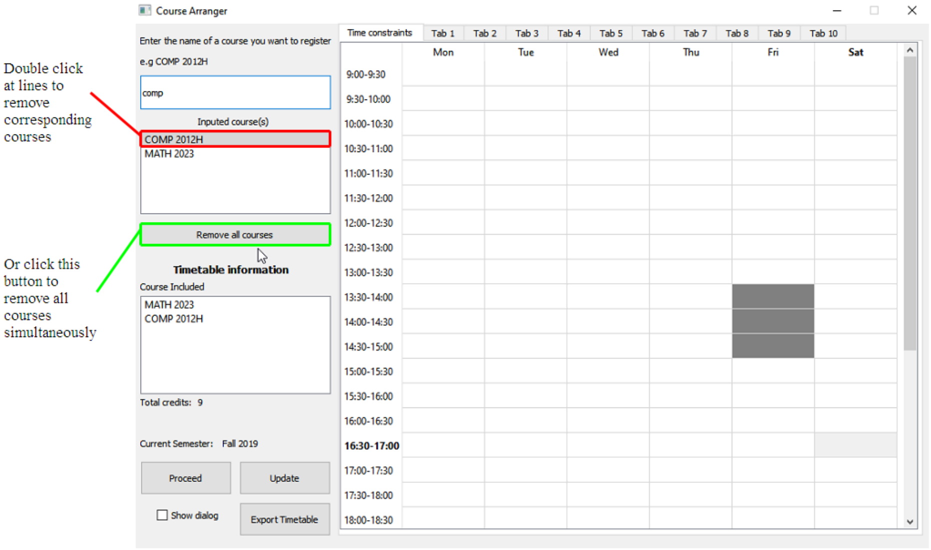
MATH 2023
COMP 2012H

Total credits: 9

Current Semester: Fall 2019

	Mon	Tue	Wed	Thu	Fri	Sat
9:00-9:30						
9:30-10:00						
10:00-10:30						
10:30-11:00		MATH 2023 L1		MATH 2023 L1		
11:00-11:30		Instructor: FUNG, Jimmy Chi Hung		Instructor: FUNG, Jimmy Chi Hung		
11:30-12:00		...		...	COMP 2012H A1	
12:00-12:30	MATH 2023 T1				Instructor: TSOI, Yau Chat	
12:30-13:00	Enrol: 38/40				Enrol: 67/67	
13:00-13:30						
13:30-14:00		COMP 2012H L1		COMP 2012H L1	Occupied	
14:00-14:30		Instructor: TSOI, Yau Chat		Instructor: TSOI, Yau Chat		
14:30-15:00		Enrol: 67/67		Enrol: 67/67		
15:00-15:30						
15:30-16:00						
16:00-16:30						
16:30-17:00						
17:00-17:30						
17:30-18:00						
18:00-18:30						

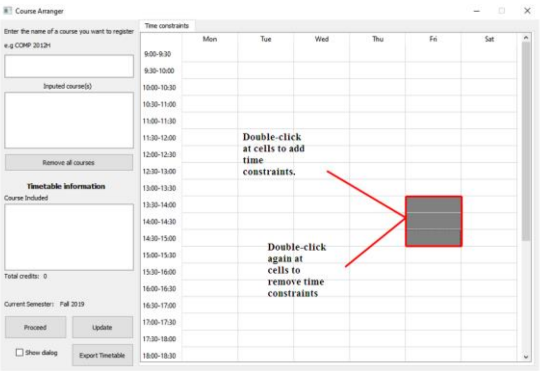
2. Remove courses:
   - Double click at specific course line to remove the course from the list
   - Click at the “Remove all courses” button to remove all the courses



Double click at lines to remove corresponding courses

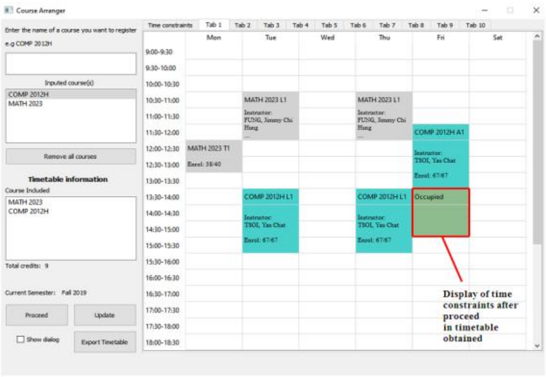
Or click this button to remove all courses simultaneously

3. Add and Remove time constraints:
   - First-time double-click at one cell of the “Time constraints” tab to shade as occupied
   - Second-time double-click at one occupied (gray) cell of the “Time constraints” tab to remove the corresponding constraint.



Double-click at cells to add time constraints.

Double-click again at cells to remove time constraints



Display of time constraints after proceed in timetable obtained

MATH 2023 L1
Instructor:
FUNG, Jimmy Chi Hung

MATH 2023 T1
Enrol: 38/40

COMP 2012H A1
Instructor:
TSOI, Yau Chat
Enrol: 67/67

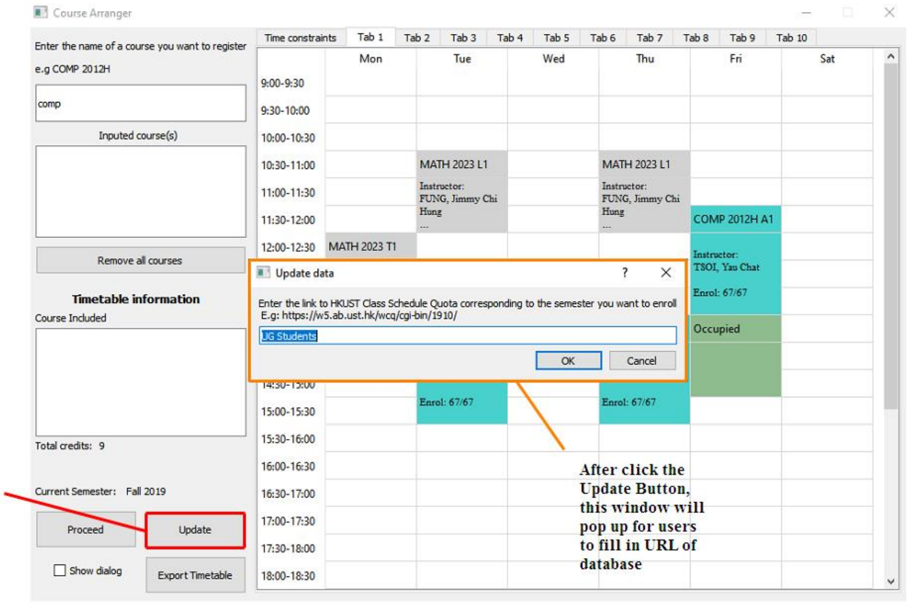
COMP 2012H L1
Instructor:
TSOI, Yau Chat
Enrol: 67/67

Occupied

Note: No need to fill in the constrained cell!

4. Update data

   Click at the “Update” button to update the data from the database (All information regarding HKUST courses and quota)



Enter the name of a course you want to register

e.g COMP 2012H

comp

Inputed course(s)

Timetable information

Course Included

Total credits: 9

Current Semester: Fall 2019

MATH 2023 L1
Instructor:
FUNG, Jimmy Chi
Hung
...

MATH 2023 T1

COMP 2012H A1
Instructor:
TSOI, Yau Chat
Enrol: 67/67

Occupied

Enrol: 67/67

Update data

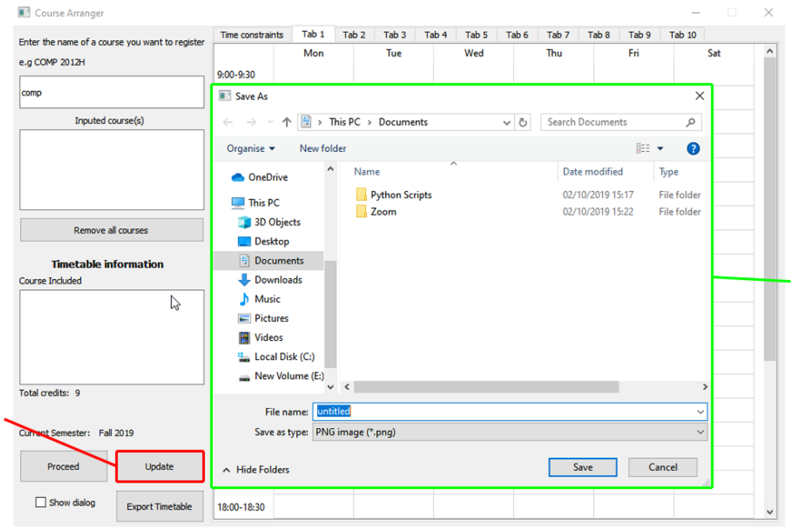
Enter the link to HKUST Class Schedule Quota corresponding to the semester you want to enroll
E.g: https://w5.ab.ust.hk/wcq/cgi-bin/1910/

UG Students

Click this button to update data from database

After click the Update Button, this window will pop up for users to fill in URL of database

5. Export results
   Export the timetable of the current tab as an image to the desired path



Enter the name of a course you want to register

e.g COMP 2012H

comp

Inputed course(s)

Timetable information

Course Included

Total credits: 9

Current Semester: Fall 2019

Name	Date modified	Type
Python Scripts	02/10/2019 15:17	File folder
Zoom	02/10/2019 15:22	File folder

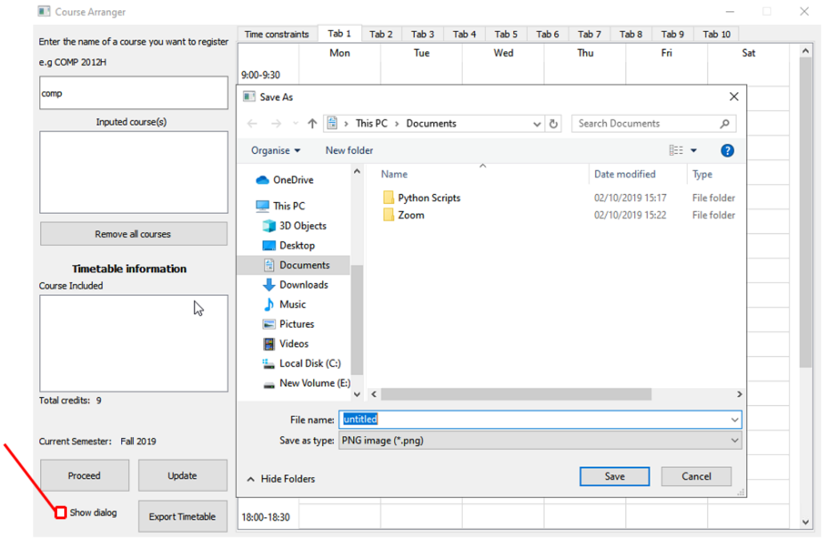
File name: untitled

Save as type: PNG image (*.png)

Click this button to export image of timetable in chosen tab

After click the Export Timetable Button, this window will pop up for users to name and choose where to save the exported image

6. Warning for confirmation
   Tick at “Show dialog” box to allow warning for double-checking when taking a new manipulation



comp

Total credits: 9

Current Semester: Fall 2019

Name	Date modified	Type
Python Scripts	02/10/2019 15:17	File folder
Zoom	02/10/2019 15:22	File folder

File name: untitled

Save as type: PNG image (*.png)

Tick this
box to see
warning for
double
checking
of actions

Main implementation:

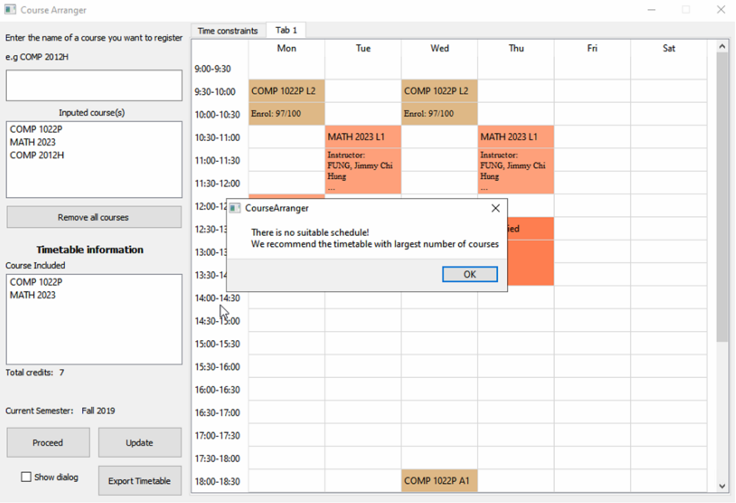
Step 1: Check whether the "Current Semester" is the upcoming semester or not. If not, firstly update the data from the database as shown in *Feature* 4.

Step 2: Add all the courses I wanted to take in the upcoming semester as shown in *Feature* 1 and 2.

Step 3: Add or remove time constraints in the "Time constraints" tab as shown in *Feature* 3.

Step 4: Click "Proceed" button to generate results. Subsequently, either one of the two situations will happen:

- The timetables obtained will contain all input courses as desired
- Due to overlapping, the timetables contained the highest possible total number of courses will be displayed.



Inputed course(s): COMP 1022P, MATH 2023, COMP 2012H

Course Included: COMP 1022P, MATH 2023

Total credits: 7

Current Semester: Fall 2019

	Mon	Tue	Wed	Thu	Fri	Sat
9:00-9:30						
9:30-10:00	COMP 1022P L2		COMP 1022P L2			
10:00-10:30	Enrol: 97/100		Enrol: 97/100			
10:30-11:00		MATH 2023 L1		MATH 2023 L1		
11:00-11:30		Instructor: FUNG, Jimmy Chi Hung ...		Instructor: FUNG, Jimmy Chi Hung ...		
11:30-12:00						
12:00-12						
12:30-13				ied		
13:00-13						
13:30-14						
14:00-14:30						
14:30-15:00						
15:00-15:30						
15:30-16:00						
16:00-16:30						
16:30-17:00						
17:00-17:30						
17:30-18:00						
18:00-18:30			COMP 1022P A1			

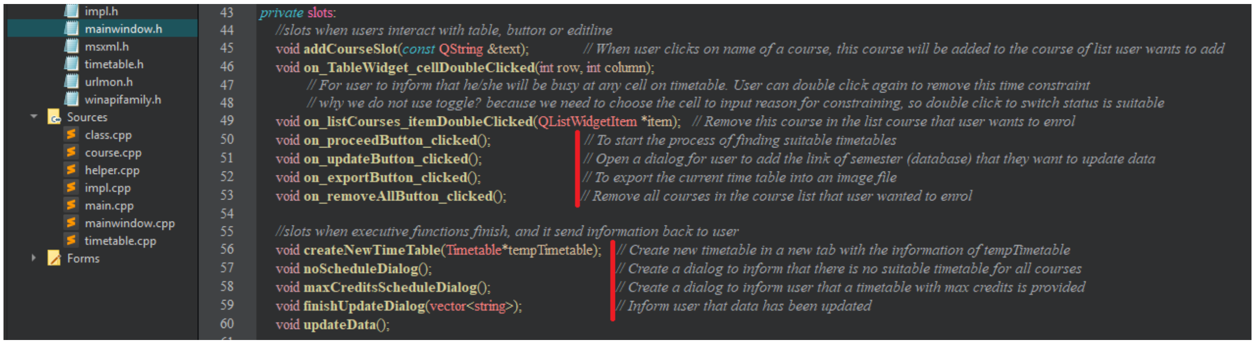
There is no suitable schedule!
We recommend the timetable with largest number of courses

Note: Users can continue to modify the input courses list and time constraints tab by following the above procedure until finding the most proper timetable.

Conclusion

Making use of Qt Creator and Urlmon library, Course Arranger is able to help HKUST students retrieve necessary data from the database - the HKUST Class Schedule and Quota, and subsequently arrange proper timetables for them based on the list of courses they want to take in the upcoming semester along with some desired time constraints. Utilizing every feature such as adding courses, time constraints… of Course Arranger, users can have two possible types of outcome: at most 50 timetables of exactly what they wish if there is no overlapping occurred; otherwise a timetable that has the highest total credits.

Appendix

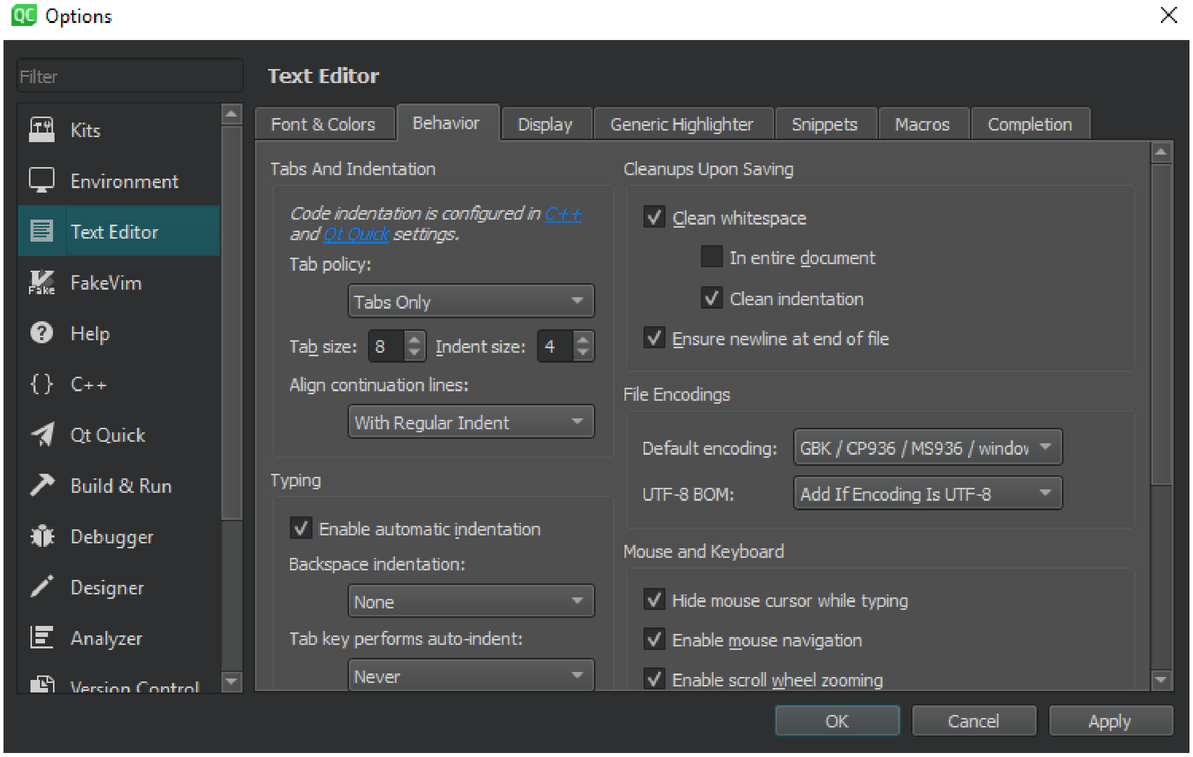


```
private slots:
    //slots when users interact with table, button or editline
    void addCourseSlot(const QString &text);            // When user clicks on name of a course, this course will be added to the course of list user wants to add
    void on_TableWidget_cellDoubleClicked(int row, int column);
            // For user to inform that he/she will be busy at any cell on timetable. User can double click again to remove this time constraint
            // why we do not use toggle? because we need to choose the cell to input reason for constraining, so double click to switch status is suitable
    void on_listCourses_itemDoubleClicked(QListWidgetItem *item);   // Remove this course in the list course that user wants to enrol
    void on_proceedButton_clicked();                    // To start the process of finding suitable timetables
    void on_updateButton_clicked();                     // Open a dialog for user to add the link of semester (database) that they want to update data
    void on_exportButton_clicked();                     // To export the current time table into an image file
    void on_removeAllButton_clicked();                  // Remove all courses in the course list that user wanted to enrol

    //slots when executive functions finish, and it send information back to user
    void createNewTimeTable(Timetable*tempTimetable);   // Create new timetable in a new tab with the information of tempTimetable
    void noScheduleDialog();                            // Create a dialog to inform that there is no suitable timetable for all courses
    void maxCreditsScheduleDialog();                    // Create a dialog to inform user that a timetable with max credits is provided
    void finishUpdateDialog(vector<string>);            // Inform user that data has been updated
    void updateData();
```

Because Qt makes some keywords bold which is different from almost other text editors, in order to see comments aligned as in the attached image, users should open source and header files by

Qt Creator within the following setup for the editor (tab size = 8, indent size = 4)



Text Editor

Tabs And Indentation

Code indentation is configured in C++ and Qt Quick settings.

Tab policy: Tabs Only

Tab size: 8 Indent size: 4

Align continuation lines: With Regular Indent

Typing

☑ Enable automatic indentation

Backspace indentation: None

Tab key performs auto-indent: Never

Cleanups Upon Saving

☑ Clean whitespace

□ In entire document

☑ Clean indentation

☑ Ensure newline at end of file

File Encodings

Default encoding: GBK / CP936 / MS936 / windov

UTF-8 BOM: Add If Encoding Is UTF-8

Mouse and Keyboard

☑ Hide mouse cursor while typing

☑ Enable mouse navigation

☑ Enable scroll wheel zooming