Class Schedule Planner (Project Arisu)

By

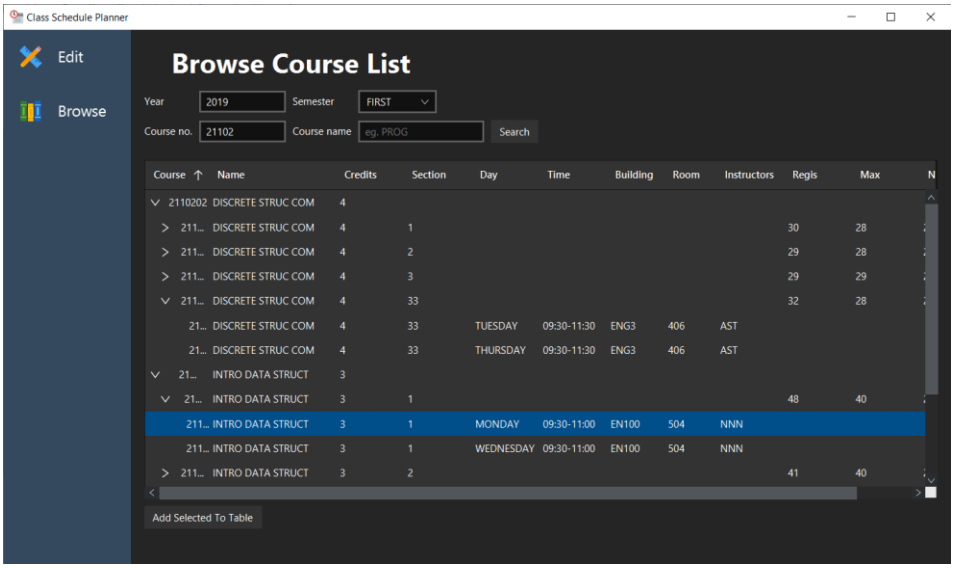
Pavat Lertpiromlak 6130396221

2110215 Programming Methodology 1
Semester 1 Academic Year 2019
Faculty of Engineering, Chulalongkorn University

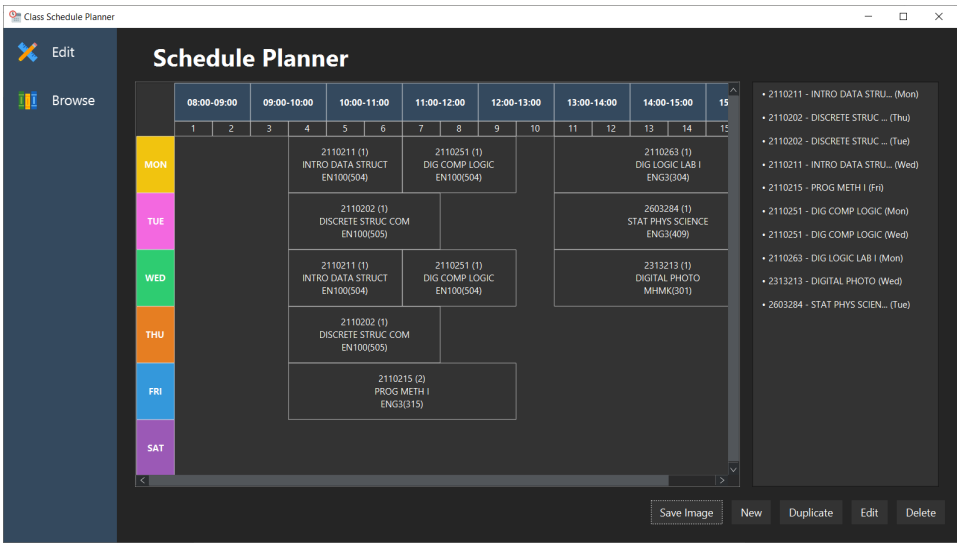
1. User Manual

Introduction

Class schedule planner is an application to search/browse for courses in reg.chula.ac.th database and make a class schedule from either user input or data from reg.chula.ac.th.

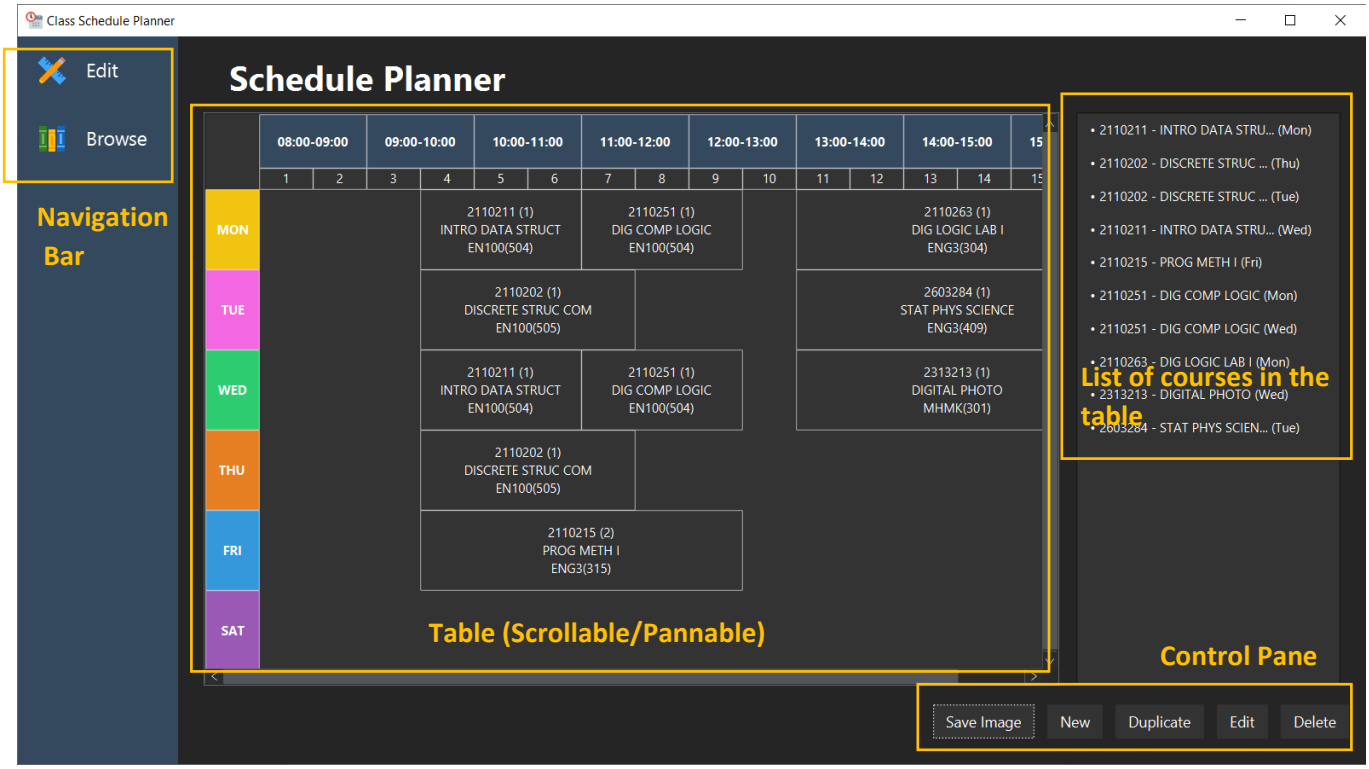


Program has 2 parts, first part is an edit function, second part is a search/browse function. You can navigate between these functionalities by clicking on navigation pane on the left side.



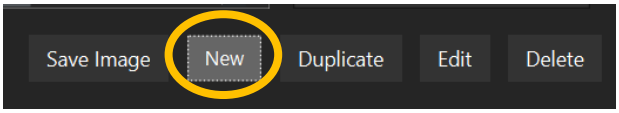
Edit (Schedule Planner)

Image below describe the components of this section .

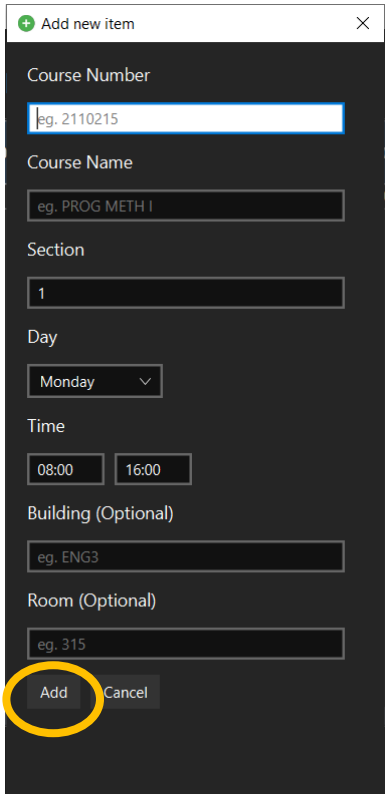


- Adding a course to the table

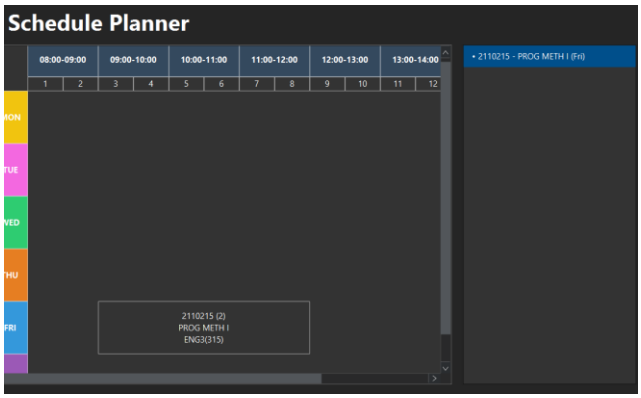
1. Click *New* Button on the Control Pane



2. A new dialog will appear, fill out an information about the course you wish to add to the table. (Note: Minimum time is 08:00 and maximum is 16:00) then click *Add*



3. If there aren't any problems with your data, then a new course will appear on the table and the list

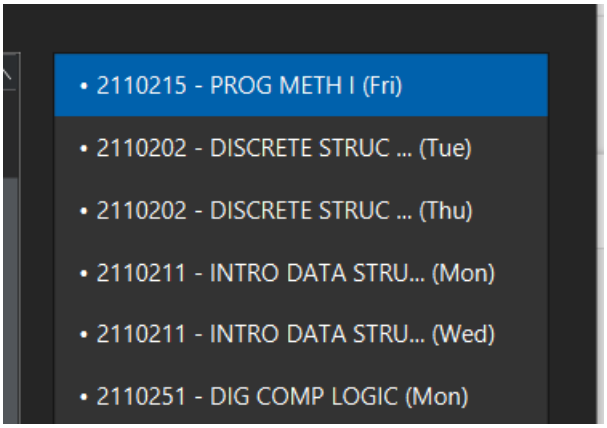


4. If there are problems with your data, then alert will appear with an error message. You must correct it before it can be added to the table. Some problems can be the following.

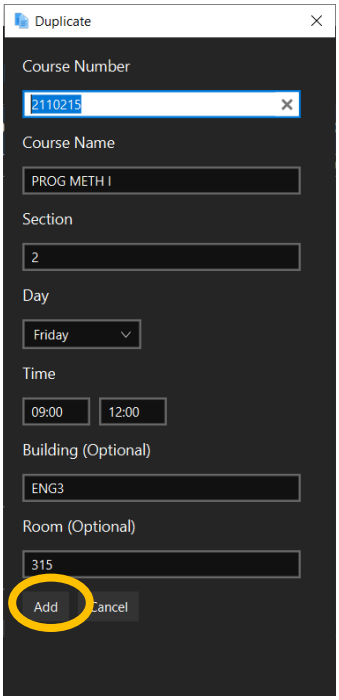
- Time conflict with existing course in the table.
- Invalid course number (grammatically).
- Time not in range (inclusive) of 08:00 – 16:00.

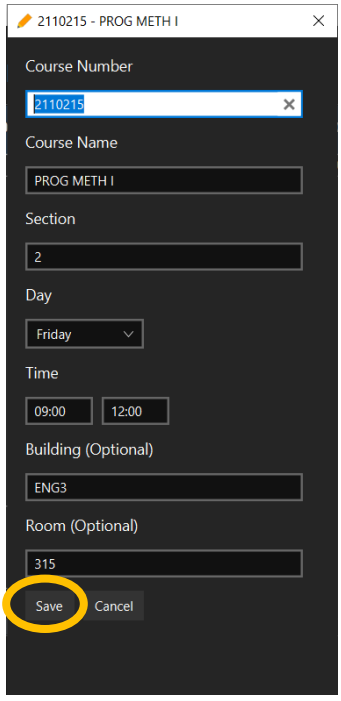
• Edit/Duplicate

1. From the right side, select a course you wish to edit or duplicate.
2. Press *Edit* or *Duplicate* to begin a process, a similar dialog when you click *Add* will appear. (Notice the different in dialog title)



3. Edit data as you want, then press *Add* or *Save*. If there is nothing problematic, then it will be saved or duplicated.
4. Possible error messages are same as Add dialog.



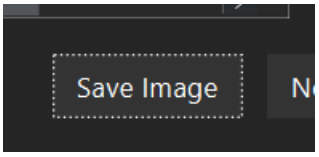


• Delete

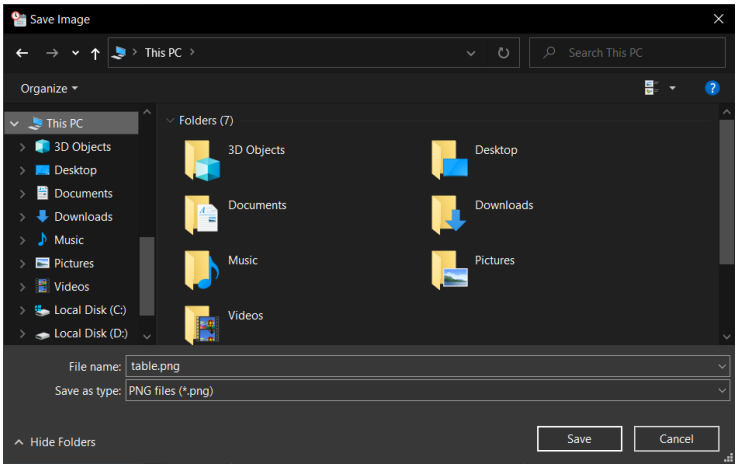
Jelect a course on the right side then click *Delete.* It will be disappeared from both the table and the list (Note: This action is irreversible!).

• Save as Image

1. Click *Save Image*



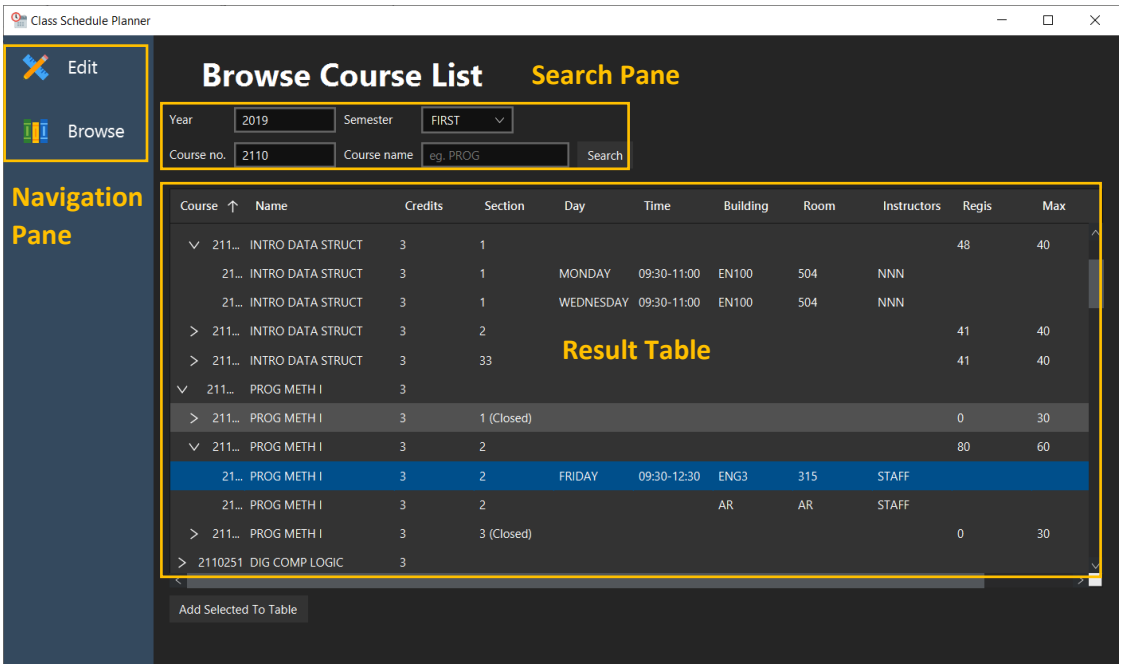
2. A save dialog will appear, choose save location and file name.



3. After that, an image file will be saved at the specified location in .png extension.

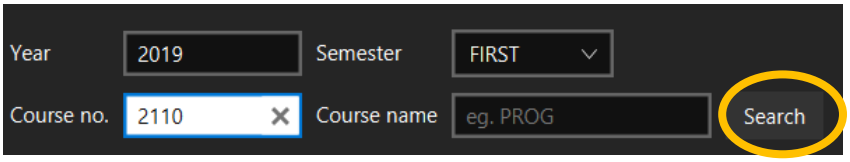
Browse (Browse Course List)

Image below describe the components of this section.

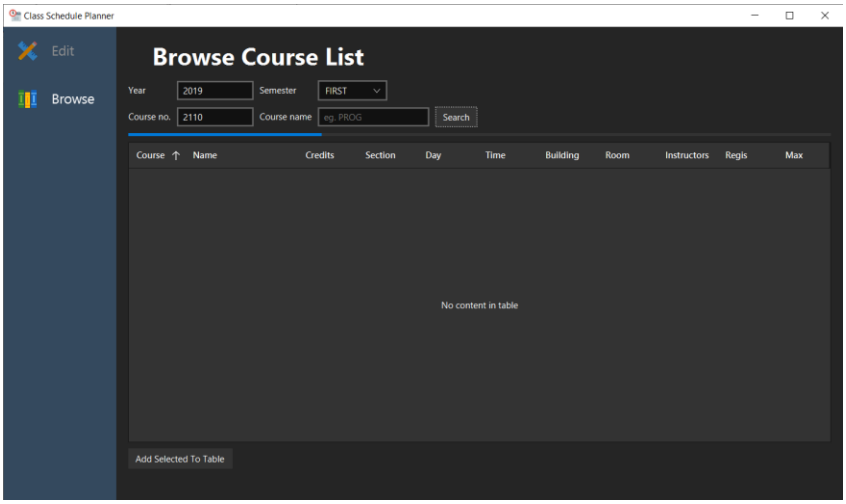


• Searching (Required an Internet Connection)

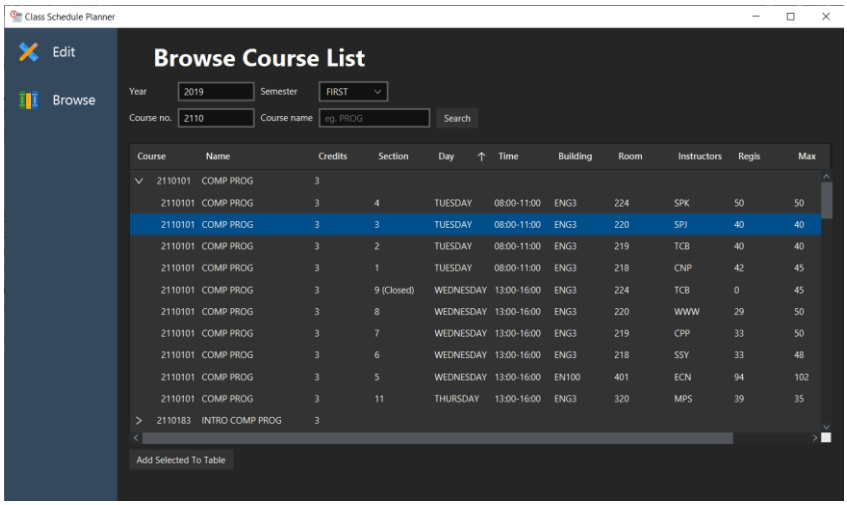
1. Type your search keywords to the search pane. Note that academic year must be in the form of Common Era (Not Buddhist era).

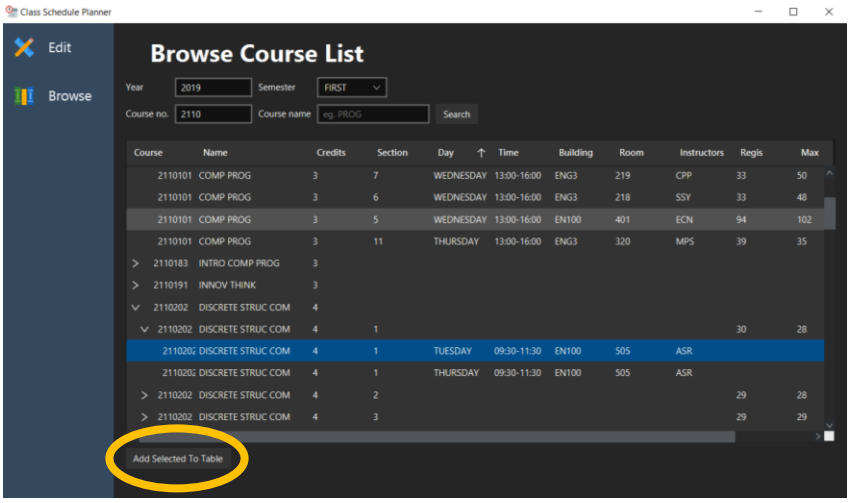


2. Press *Search* then wait for result to come up.

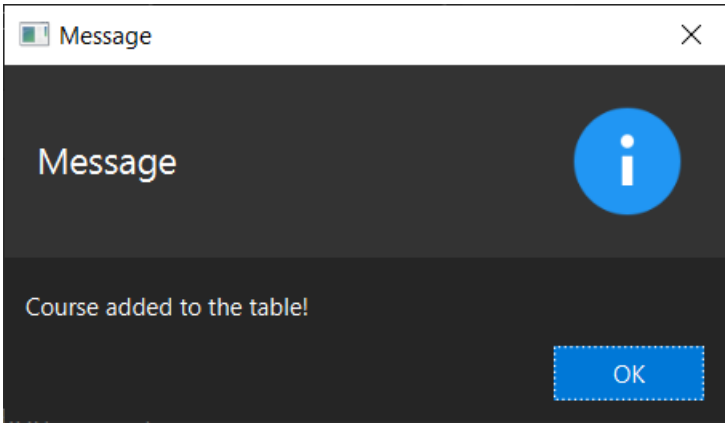


3. After that, the result will be display in the table with nested cell represent each section of a course. If a section has multiple time schedule, then it will also display in a nested format.

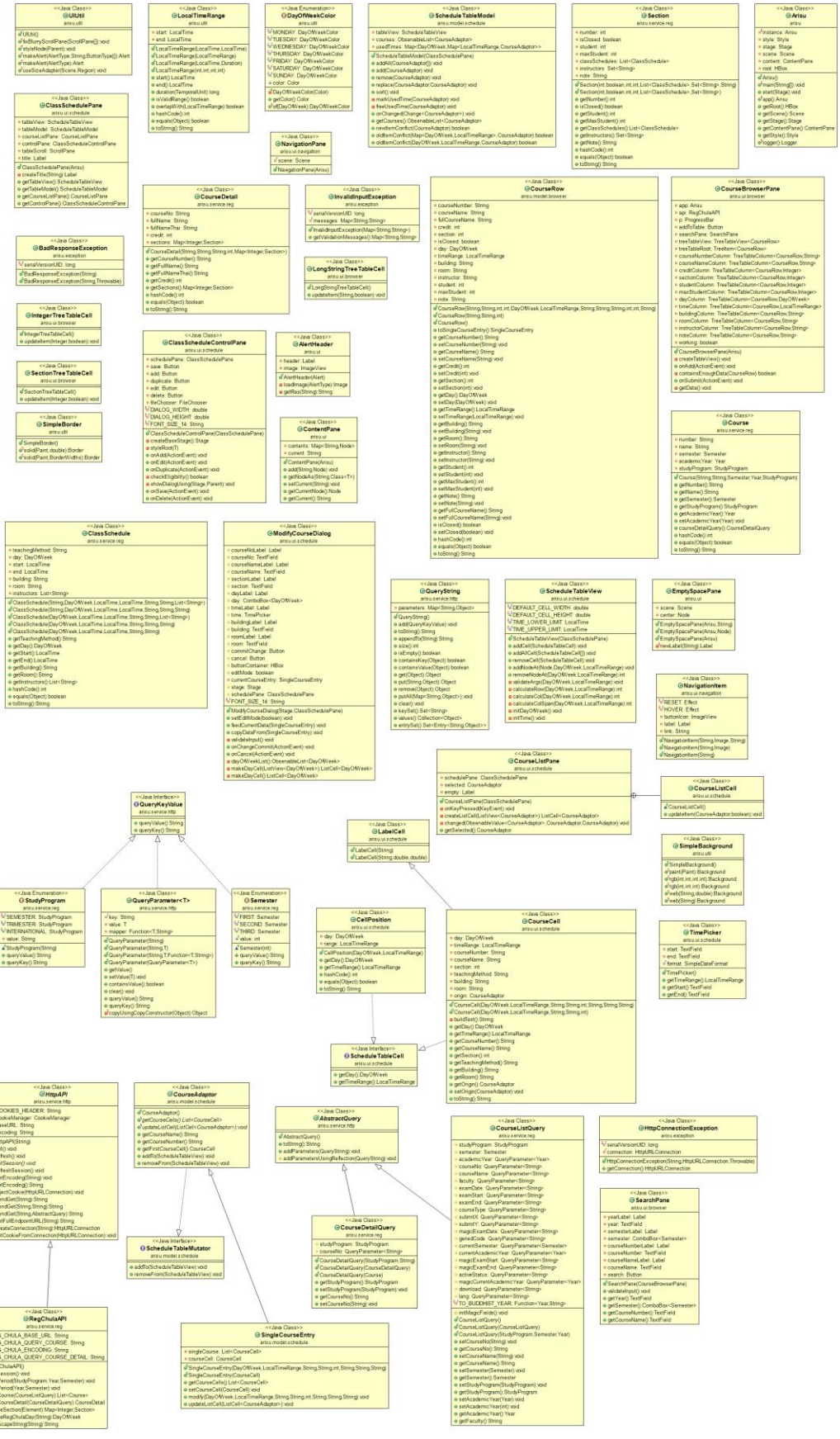




4. You can click *Add Selected to Table* to add selected item to the table. Selected item must contain at least a day and a time of that course. After that if there aren't any problems with the selected e.g. time conflict, then it will show a dialog confirming that a course was added to the table.



2. Implementation Details



** Noted that Access Modifier Notations can be listed below*

+ (public)
 # (protected)
- (private)
Underlined (static)
Italic (abstract)

Package arisu

Class Arisu extends Application

This class represents the main entry of the application.

Field

- Arisu instance	Instance of the application.
- jfextras.styles.jmetro.Style style	Represent the color scheme of the JMetro theme, set to Style.DARK.
- Stage stage	Store the primary stage of the application.
- Scene scene	Store the scene of the application.
- ContentPane content	Store the ContentPane component.
- HBox root	Root node of the application.

Methods

+ void main(String[] args)	Launch JavaFX application.
+ void start(Stage primaryStage)	Main entry point of JavaFX application. - Instantiate root (HBox), ContentPane, ClassSchedulePane, CourseBrowserPane - Instantiate NavigationPane then set the component to be shown for each link - Style root using UIUtil.styleNode(root) - Add components to the root - Create new scene and stage then show the stage to user
+ Arisu instance()	Get instance of the application.
+ Logger logger()	Get the application logger.
Getter for every non-static fields	

Package arisu.exception

This package contains all exceptions.

- Class BadResponseException extends Exception

This exception is thrown when http response couldn't be understand or parsed by the application.

- Class HttpConnectionException extends Exception

This exception is thrown when there are problems with http connection to some http endpoint.

- Class InvalidInputException extends Exception

This exception is thrown when user inputs are invalid.

Package arisu.model

This package contains data model for the UI of the application.

Package arisu.model.browser

Class CourseRow

Represent data model of the table in the browse/search function.

Field

- String courseNumber	Course number in 7 digits e.g. "2110215".
- String courseName	Short course name e.g. "PROG METH I".
- String fullCourseName	Full course name e.g. "PROGRAMMING METHODOLOGY I".
- int credit	Credit granted for this course e.g. 3.
- int section	What is the section of this course?
- boolean isClosed	Is this section closed?
- DayOfWeek day	Day of week enum (MONDAY to SUNDAY).
- LocalTimeRange timeRange	Represent start time and end time for this course for specific day of week.
- String building	Building
- String room	Room
- String instructor	All of Instructor

- int student	Currently registered student of this section.
- int maxStudent	Maximum student for this section.
- String note	Note column, there might be some details related to this section, e.g. GenED

Constructor

+ CourseRow(String courseNumber, String courseName, int credit, int section, DayOfWeek day, LocalTimeRange timeRange, String building, String room, String instructor, int student, int maxStudent, String note)	Initialize each field from constructor parameters.
+ CourseRow(String courseNumber, String courseName, int credit)	Initialize each field from constructor parameters.
+ CourseRow()	Empty constructor, Will use default value for each field.

Method

+ SingleCourseEntry toSingleCourseEntry()	Create SingleCourseEntry from field values of this object
Getter/Setter for every fields	
+ int hashCode()	Generate hash code for this object using all data field of this object.
+ boolean equals(Object obj)	Check if another object equals itself. It checks if for all fields does it equals?
+ String toString()	Generate string representation of this object.

Package arisu.model.schedule

Abstract class CourseAdapter implements ScheduleTableMutator

Represent a course data that can be added and remove from/to the list and the table

Method

+ List<CourseCell> getCourseCells()	Return all possible cells to be added to the table. (Current implementation (SingleCourseEntry) only has one cell)
+ void updateListCell(ListCell<CourseAdapter> listCell);	Update the data in the ListView.
+ String getCourseName()	Get course name of the first course cell.
+ String getCourseNumber()	Get course number of the first course cell.

+ CourseCell getFirstCourseCell()	Get first CourseCell from the list of getCourseCells().

Class ScheduleTableModel

Represent data model on the schedule table.

Field

- ScheduleTableView tableView	Component for rendering a table.
- ObservableList<CourseAdapter> courses	List of courses in the table.
- Map<DayOfWeek, Map<LocalTimeRange, CourseAdapter>> usedTimes	Store data whether this day and time has been used by something or not?

Constructor

+ ScheduleTableModel(ClassSchedulePane schedulePane)	Initialize all fields. Add the listener to courses list to observe change in the list.

Method

+ void addAll(CourseAdapter …courses)	Variadic arguments version of add(CourseAdapter courseAdapter).
+ void add(CourseAdapter courseAdapter)	Add CourseAdapter object to the list. Also call markUsedTime(courseAdapter) to mark that day and time has been used.
+ void remove(CourseAdapter courseAdapter)	Remove CourseAdapter object from the list. Also call freeUsedTime(courseAdapter) to unmark that day and time that had been used.
+ void replace(CourseAdapter oldVal, CourseAdapter newVal)	Replace old item with new one in the same index.
+ void sort()	Sort the item in the list.
- void markUsedTime(CourseAdapter courseAdapter)	Mark that specified courseAdapter has used some day and time to be display on the table.
- void freeUsedTime(CourseAdapter courseAdapter)	Undo the markUsedTime(courseAdapter) method.
+ void onChanged(ListChangeListener.Change<? extends CourseAdapter> change)	Listen for change in the list. If item was removed then call courseAdapter.removeFrom(tableView) else call courseAdapter.addTo(tableView).
+ ObservableList<CourseAdapter> getCourses()	Return the list of CourseAdapter object.

+ boolean newItemConflict(CourseAdapter input)	Check for time conflict with existing courseAdaptor
+ boolean oldItemConflict(Map<DayOfWeek, LocalTimeRange> times, CourseAdapter except)	Check for time conflict with existing courseAdapter, this method check using all of days and times in the map with exception for some courseAdapter object. Use when edit the list.
+ boolean oldItemConflict(DayOfWeek dayOfWeek, LocalTimeRange timeRange, CourseAdapter except)	Check for time conflict with existing courseAdapter, this method check using specific day and time with exception for some courseAdapter object. Use when edit the list.

Interface ScheduleTableMutator

Represent an object that can mutate the schedule table (By adding to the table or remove from the table).

Method

void addTo(ScheduleTableView tableView)	Called when you want to add that object presentation to the schedule table.
void removeFrom(ScheduleTableView tableView)	Called when you want to remove that object presentation from the schedule table.

Class SingleCourseEntry extends CourseAdapter

Current and only single course, single day and time implementation of CourseAdaptor

Field

- List<CourseCell> singleCourse	List of single CourseCell object
- CourseCell courseCell	CourseCell object

Constructor

+ SingleCourseEntry(DayOfWeek day, LocalTimeRange timeRange, String courseNumber, String courseName, int section, String building, String room, String teachingMethod)	Construct CourseCell object from these parameters then delegate to SingleCourseEntry(courseCell) constructor.

+ SingleCourseEntry(CourseCell courseCell)	Assign courseCell to field courseCell then create a list of single element of courseCell then assign to singleCourse. Call setOrigin(this) on courseCell to mark the owner of modification.

Method

+ List<CourseCell> getCourseCells()	Return singleCourse field
+ void setCourseCell(CourseCell courseCell)	Set courseCell to the parameter then call setOrigin(this) on courseCell to mark the owner of modification.
+ void modify(DayOfWeek day, LocalTimeRange timeRange, String courseNumber, String courseName, int section, String building, String room, String teachingMethod)	Set courseCell to the new CourseCell constructed from method parameters. Then call setOrigin(this) on courseCell to mark the owner of modification.
+ void updateListCell(ListCell<CourseAdapter> listCell)	Set the content of ListCell when this is being displayed on the table.

Package arisu.service

Package arisu.service.http

Class related to http communication

Abstract class AbstractQuery

Represent a query that can be turn into a query string.

Method

+ String toString()	Get query string of this query.
+ void addParameters(QueryString qs)	Override this method to add parameter to query string builder.
# void addParametersUsingReflection(QueryString qs)	Use reflection to get all field that is an instance of QueryKeyValue then get the key and value to build the query string.

Abstract class HttpAPI

A base class for helping in Http API implementation.

Field

final String COOKIE_HEADER	Http header for cookie, set to “Set-Cookie”
# CookieManager cookieManager	Cookie manager for this Http api.
# String baseURL	Base URL for this Http API.
# String encoding	Encoding of Http response.

Constructor

+ HttpAPI(String baseURL)	Construct Http API client with specific base url.

Method

+ void init()	Initialize the API.
+ void refresh()	Refresh the API.
+ void initSession()	Initiate Http session connection to the API. Required by stateful API.
+ void refreshSession()	Refresh http session.
+ void setEncoding(String encoding)	Set encoding to use in response processing.
+ String getEncoding()	Get encoding.
# void injectCookie(HttpURLConnection connection)	Inject cookie into HttpURLConnection
+ String sendGet(String endpoint)	Send get request to some endpoint.
+ String sendGet(String endpoint, String queryString)	Send get request to some endpoint with query string.
+ String sendGet(String endpoint, AbstractQuery query)	Send get request to some endpoint with AbstractQuery object.
# String getFullEndpointURL(String endpoint)	Create full Http url to send the request.
# HttpURLConnection createConnection(String endpoint)	Create Http connection
# void initCookieFromConnection(HttpURLConnection connection)	Retrieve cookie from the server.

Interface QueryKeyValue

Represent pair of query string key value.

Method

+ String queryKey()	Get query string key.
+ String queryValue()	Get query string value.

Class QueryParameter<T> implements QueryKeyValue

Query parameter wrapping any object that can be represent as QueryKeyValue.

Field

- final String key	Key of this parameter.
- T value	Value of this parameter.
- Function<T, String> mapper	Optional function for post processing retrieved parameter value.

Constructor

+ QueryParameter(String key)	Create QueryParameter with key.
+ QueryParameter(String key, T value)	Create QueryParameter with key and value.
+ QueryParameter(String key, T value, Function<T, String> mapper)	Create QueryParameter with key and value and mapper function.
+ QueryParameter(QueryParameter<T> qp)	Copy constructor

Method

+ T getValue()	Get value of the object.
+ void setValue(T value)	Set value of the object.
+ boolean containsValue()	Check whether value is not null.
+ void clear()	Set value to null.
+ String queryValue()	Get query value.
+ String queryKey()	Get query key.
- Object copyUsingCopyConstructor(Object o)	Utility method for copying object using its copy constructor if possible.

Class QueryString implements Map<String, Object>

Query string builder.

Field

- Map<String, Object> parameters	Key value map of query string.

Constructor

+ QueryString()	Instantiate parameters using LinkedHashMap to maintain insertion order.

Method

+ void add(QueryKeyValue query)	Add QueryKeyValue to query builder.
+ String toString()	Convert into query string.
+ String appendTo(String another)	Join another and this.toString() using “?”.
Other methods from Map<String,Object>	Delegate a call to the parameters instance.

Package arisu.service.reg

Reg.chula.ac.th API

Class ClassSchedule

Represent a class schedule on specific day and time range (Note: some fields are currently unused by the GUI)

Field

- String teachingMethod	Teaching method e.g. “LECT”, “LAB”
- DayOfWeek day	Day of this schedule
- LocalTime start	Starting time
- LocalTime end	Ending time
- String building	Building
- String room	Room
- List<String> instructors	List of instructors

Constructor

+ ClassSchedule(String teachingMethod, DayOfWeek day, LocalTime start, LocalTime end, String building, String room, List<String> instructors)	Initialize field according to parameters given.

+ ClassSchedule(String teachingMethod, DayOfWeek day, LocalTime start, LocalTime end, String building, String room, String instructor)	Delegate to first constructor. - set instructrs to List.of(instructor)
+ ClassSchedule(DayOfWeek day, LocalTime start, LocalTime end, String building, String room, List<String> instructors)	Delegate to first constructor. - set teachingMethod to “LECT”.
+ ClassSchedule(DayOfWeek day, LocalTime start, LocalTime end, String building, String room, String instructor)	Delegate to first constructor. Missing parameters is same as second and third
+ ClassSchedule(DayOfWeek day, LocalTime start, LocalTime end, String building, String room)	Delegate to first constructor. - set instructrs to List.of(“STAFF”) - set teachingMethod to “LECT”.

Method

Getter for every fields	
+ int hashCode()	Get hash code.
+ boolean equals(Object obj)	Check if another object equals to this.
+ String toString()	String representation of this object.

Class Course

Represent brief detail of course for the basis step of querying (As you can see when you firstly click search on reg.chula).

Field

- String number	Course number
- String name	Course name (short name)
- Semester semester	Semester of academic year
- Year academicYear	Academic year
- StudyProgram studyProgram	How semester is divived?

Constructor

+ Course(String number, String name, Semester semester, Year academicYear, StudyProgram studyProgram)	Initialize field according to parameters given.

Method

Getter for every fields	
+ int hashCode()	Get hash code.
+ boolean equals(Object obj)	Check if another object equals to this.

+ String toString()	String representation of this object.

Class CourseDetail

Represent more details about a course (As you can see when you click course id from the search result on reg.chula).

Field

- String courseNo	Course number
- String fullName	Course name (full name)
- String fullNameThai	Course name (full name in Thai language)
- int credit	Credit granted for this course
- Map<Integer, Section> sections;	Map of section number and section details

Constructor

+ Course(String number, String name, Semester semester, Year academicYear, StudyProgram studyProgram)	Initialize field according to parameters given.

Method

Getter for every fields	
+ int hashCode()	Get hash code.
+ boolean equals(Object obj)	Check if another object equals to this.
+ String toString()	String representation of this object.

Class CourseDetailQuery extends AbstractQuery

Query to get list of courses.

Field

# StudyProgram studyProgram;	Study program
# QueryParameter<String> courseNo;	Course number

Constructor

+ CourseDetailQuery(StudyProgram studyProgram, String courseNo)	Initialize object using study program and course number.
+ CourseDetailQuery(CourseDetailQuery q)	Copy constructor
+ CourseDetailQuery(Course course)	Initialize object using course object.

Method

Getter for every field	

Class CourseListQuery extends AbstractQuery

Query to get details of course.

Field

# StudyProgram studyProgram	Study program
# Semester semester	Semester
# QueryParameter<Year> academicYear	Academic year
# QueryParameter<String> courseNo	Course number
# QueryParameter<String> courseName	Course name
# QueryParameter<String> faculty	Faculty
# QueryParameter<String> examDate	Exam date
# QueryParameter<String> examStart	Exam start time
# QueryParameter<String> examEnd	Exam end time
# QueryParameter<String> courseType	Couse type
# QueryParameter<String> submitX	Magic/Unknown parameter
# QueryParameter<String> submitY	
# QueryParameter<String> magicExamDate	
# QueryParameter<String> genedCode	
# QueryParameter<Semester> currentSemester	
# QueryParameter<Year> currentAcademicYear	
# QueryParameter<String> magicExamStart	
# QueryParameter<String> magicExamEnd	
# QueryParameter<String> activeStatus	
# QueryParameter<Year> magicCurrentAcademicYear	
# QueryParameter<String> download	
# QueryParameter<String> lang	
- final Function<Year, String> TO_BUDDHIST_YEAR	year -> Integer.toString(year.getValue() + 543)

Constructor

+ CourseListQuery()	Initialize with default value.
+ CourseListQuery(CourseListQuery q)	Copy constructor

+ CourseListQuery(StudyProgram studyProgram, Semester semester, Year academicYear)	Initialize with three commonly used parameters.

Method

+ void setCourseNo(String courseNo)	Set course number. Then extract faculty code and assign to faculty field.
Getter/Setter for every other non-magic fields	

Class RegChulaAPI extends HttpAPI

(Unofficial) API for querying about course from reg.chula.ac.th

Field

final String REG_CHULA_BASE_URL	"https://cas.reg.chula.ac.th/"
final String REG_CHULA_QUERY_COURSE	"/servlet/com.dtm.chula.cs.servlet.QueryCourseScheduleNew.CourseListNewServlet"
final String REG_CHULA_ENCODING	"x-windows-874"
final String REG_CHULA_QUERY_COURSE_DETAIL	"/servlet/com.dtm.chula.cs.servlet.QueryCourseScheduleNew.CourseScheduleDtlNewServlet"

Constructor

+ RegChulaAPI()	Call super constructor to set base url. Then set the encoding to REG_CHULA_ENCODING

Method

+ void initSession()	Get cookie from the server.
+ void setPeriod(StudyProgram studyProgram, Year academicYear, Semester semester)	Change current study program, academic year and semester acknowledgement of the server.
+ void setPeriod(Year academicYear, Semester semester)	Another method of setPeriod() assuming semester study program.
+ List<Course> getCourse(CourseListQuery query)	Get list of courses.
+ CourseDetail getCourseDetail(CourseDetailQuery query)	Get course detail

- Map<Integer, Section> parseSection(Element sectionsElement)	Parse section html data into map of section number and section details.
- DayOfWeek parseRegChulaDay(String day)	Parse day of week from reg.chula into DayOfWeek object
- String unescapeString(String htmlEscapedString)	Unescape html entites. And remove \u00A0 character from string.

Class Section

Represent details of each section

Field

- int number	Section number
- boolean isClosed	Is this section closed?
- int student	No. of student registered for this course
- int maxStudent	Maximum no. of student for this course
- List<ClassSchedule> classSchedules	List of all class schedule in the week
- Set<String> instructors;	All instructors for this section
- String note;	Note about this section

Constructor

+ Course(String number, String name, Semester semester, Year academicYear, StudyProgram studyProgram)	Initialize field according to parameters given.

Method

Getter for every fields	
+ int hashCode()	Get hash code.
+ boolean equals(Object obj)	Check if another object equals to this.
+ String toString()	String representation of this object.

Enum Semester implements QueryKeyValue

Enum value

FIRST	First semester

SECOND	Second semester
THIRD	Third semester (aka Summer)

Field

final int value	Ordinal number for semester

Constructor

Semester(int value)	Create Enum with number representation for that semester.

Method

+ String queryKey()	Always return “semester”.
+ String queryValue()	Return value of number representation of this semester.

Enum StudyProgram implements QueryKeyValue

Enum value

SEMESTER	Two semester system. Represent by code “S”
TRIMESTER	Three semester system. Represent by code “T”
INTERNATIONAL	International semester system. Represent by code “I”

Field

- String value	String code for study program

Constructor

Semester(String value)	Create Enum with String representation for that study program.

Method

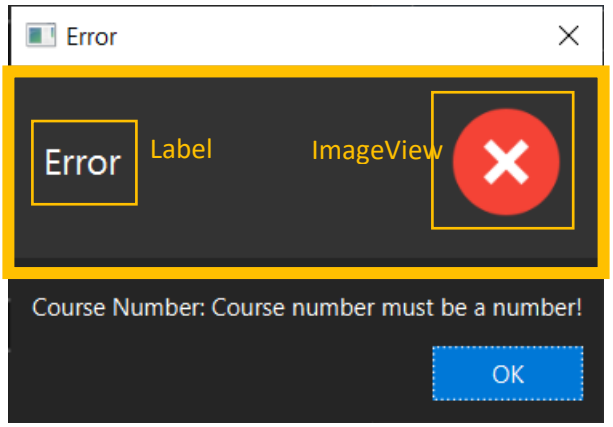
+ String queryValue()	Always return “studyProgram”.
+ String queryKey()	Return value of number representation of this semester.

Package arisu.ui

A collection of UI related class.

Class AlertHeader extends GridPane

A Customized Alert header.



Field

- Label header	Header label
- ImageView image	Icon for that type of alert

Constructor

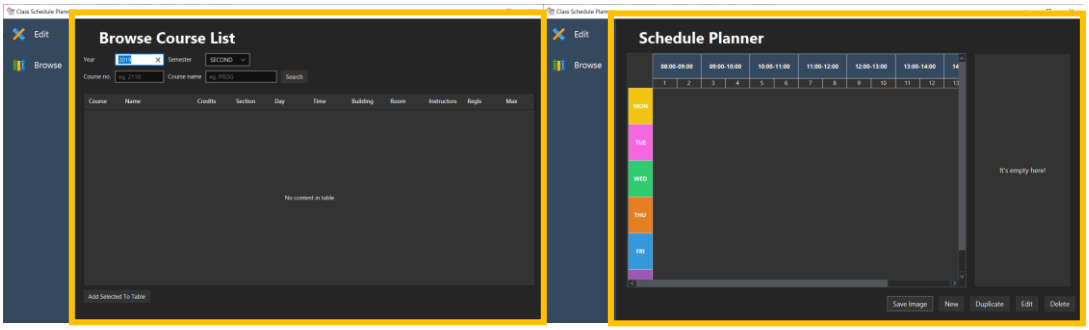
+ Alert(Alert alert)	Intialize Label and ImageView

Method

- Image loadImage(AlertType alertType)	Load image associate with given alert type.
- String getRes(String url)	Load resource from class path.

Class ContentPane extends StackPane

A StackPane that can change its content using navigation bar.



Field

- Map<String, Node> contents	Map of name and node to be displayed.

- String current	Currently selected link.

Constructor

+ ContentPane(Arisu app)	Assign contents field with instance of new HashMap.

Method

+ void add(String link, Node node)	Add node to ContentPane.
+ <T extends Node> T getNodeAs(String link, Class<T> clazz)	Get node using name and then typecast to type of that node.
+ void setCurrent(String name)	Set current node to be display by name.
+ Node getCurrentNode()	Return currently displayed node.
+ String getCurrent()	Return current name.

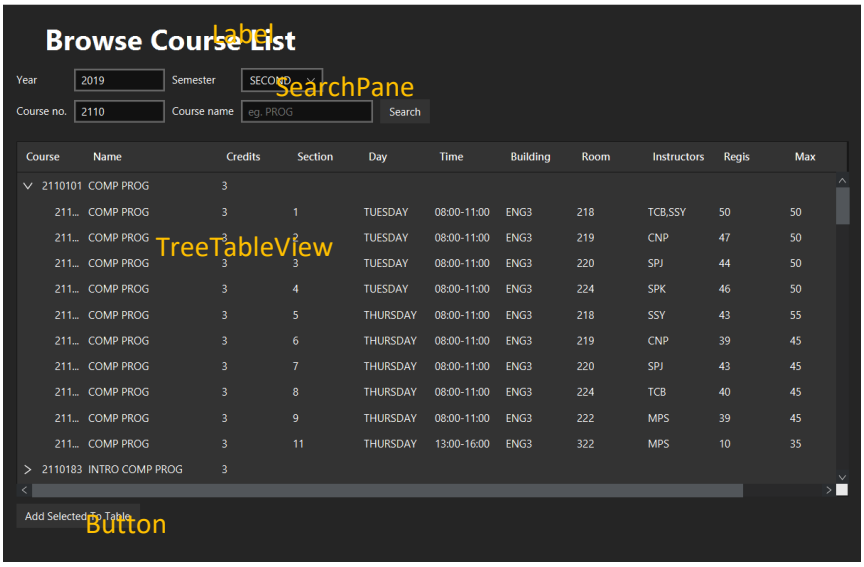
Class EmptySpacePane extends StackPane

A StackPane with label at the center (Used in early prototyping, now unused).

Package arisu.ui.browser

Components for browser/search feature.

Class CourseBrowserPane extends VBox



Field

- Arisu app	App instance
- RegChulaAPI api = new RegChulaAPI()	Reg.chula API

- ProgressBar p;	Progress bar component
- Button addToTable	Add to the table button
- SearchPane searchPane	Search pane
- TreeTableView<CourseRow> treeTableView	Tree table component
- TreeItem<CourseRow> treeTableRoot	Root of tree table
- TreeTableColumn<CourseRow, String> courseNumberColumn	Course number column
- TreeTableColumn<CourseRow, String> courseNameColumn	Course name column
- TreeTableColumn<CourseRow, Integer> creditColumn	Credit column
- TreeTableColumn<CourseRow, Integer> sectionColumn	Section column
- TreeTableColumn<CourseRow, Integer> studentColumn	Student column
- TreeTableColumn<CourseRow, Integer> maxStudentColumn	Max student column
- TreeTableColumn<CourseRow, DayOfWeek> dayColumn	Day column
- TreeTableColumn<CourseRow, LocalTimeRange> timeColumn	TimeRange column
- TreeTableColumn<CourseRow, String> buildingColumn	Building column
- TreeTableColumn<CourseRow, String> roomColumn	Room column
- TreeTableColumn<CourseRow, String> instructorColumn	Instructor column
- TreeTableColumn<CourseRow, String> noteColumn	Note column
- boolean working = false	It is currently fetching the data from server?

Method

+ CourseBrowserPane(Arisu app)	Initialize each components. - bind onAction of addToTable to onAdd method - Initialize tableView using createTableView() method
- void createTableView()	Create TreeTableView, initialize each column and set cell value, cell value factory that is appropriate for each column.
+ void onAdd(ActionEvent event)	Called when user click Add to Table button. Will attempt to add user selected row to the

	schedule table. If no problems persists, then it will display success dialog.
- boolean containsEnoughData(CourseRow courseRow)	Check if courseRow object contains enough data to be add to the schedule table.
+ void onSubmit(ActionEvent event)	Called when user click the search button. Will begin searching if working == false.
- void getData()	Retrieve data from reg.chula using input from SearchPane then add them to the TreeTableView.

Class IntegerTreeTableCell extends TreeTableCell<CourseRow, Integer>

Integer table cell that display its value if value is not -1.

Method

+ void updateItem(Integer item, boolean empty)	Call setText to item if value is not empty and not equals -1, otherwise null.

Class LongStringTreeTableCell extends TreeTableCell<CourseRow, String>

String table cell that will display a tooltip when user hover over the cell.

Method

+ void updateItem(String item, boolean empty)	Call setTooltip(item) and setText(item) if item is not empty.

Class SectionTreeTableCell extends TreeTableCell<CourseRow, Integer>

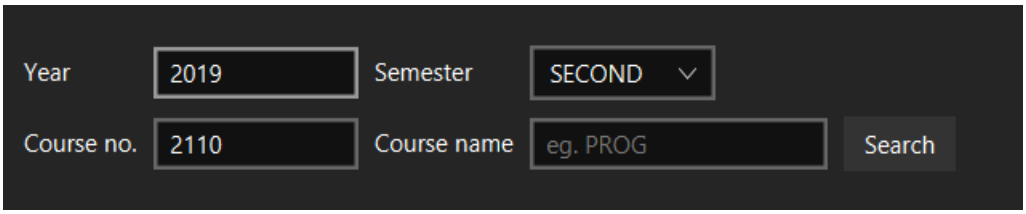
Integer table cell that display its value if value is not -1.

Method

+ void updateItem(Integer item, boolean empty)	Same as IntegerTreeTableCell.updateItem(), but if current course section associate with

	its row is closed then it will append text “(Closed)” to a section number.

Class SearchPane extends GridPane



Field

- Label yearLabel = new Label("Year")	Year label
- TextField year = new TextField("2019");	Year text field
- Label semesterLabel = new Label("Semester")	Semester label
- ComboBox<Semester> semester = new ComboBox<>(FXCollections.observableArrayList(Semester.values()))	Semester combo box
- Label courseNumberLabel = new Label("Course no.")	Course no. label
- TextField courseNumber = new TextField("")	Course no. text field
- Label courseNameLabel = new Label("Course name")	Course name label
- TextField courseName = new TextField("")	Course name text field
- Button search = new Button("Search")	Search button

Constructor

+ SearchPane(CourseBrowserPane browserPane)	Initialize each component.

Method

- void validateInput()	Validate if input is correct, otherwise throw an exception.
Getter for year, semester, courseNumber, courseName	

Package arisu.ui.navigation

Navigation bar components.

Class NavigationItem extends HBox

Field

- final Effect RESET = new ColorAdjust()	Effect when user is not hovering the button
- final Effect HOVER = new ColorAdjust(0, 0, -0.25, 0)	Effect when user is hovering the button
- ImageView buttonIcon	Button icon
- Label label	Label for button
- String link	Name used for swap between node in ContentPane.

Constructor

+ NavigationItem(String name, Image image, String link)	Initialize each component. - On mouse enter, set effect to HOVER. - On mouse exit, set effect to RESET. - On moue click, navigate to content specified in link.
+ NavigationItem(String name, Image image)	Delegate constructor - link = name.toLowerCase()

Class NavigationPane extends VBox

Container of NavigationItem

Field

- final Scene scene	scene

Constructor

- NavigationPane(Arisu app)	Initialize component - Add browse navigation item, edit navigation item - Add listener to scene height property to adapt its height to scene height.

Package arisu.ui.schedule

Components for schedule table feature.

Class CellPosition implements ScheduleTableCell

Helper object that implements ScheduleTableCell for getting data from the schedule table grid (Used in early prototyping).

Field

- DayOfWeek day;	Day
- LocalTimeRange range	Time range

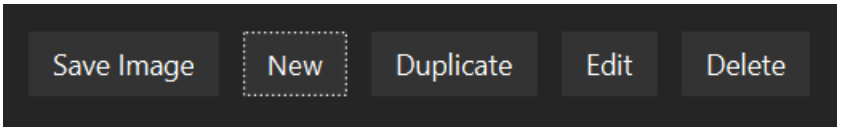
Constructor

- CellPosition(DayOfWeek day, LocalTimeRange range)	Initialize fields.

Method

Getter for every fields	
int hashCode()	Get hash code
boolean equals(Object obj)	Check if this is equals to another object.
String toString()	String representation of this object.

Class ClassScheduleControlPane extends HBox



Field

- ClassSchedulePane schedulePane	Reference to schedule pane
- Button save	Save image button
- Button add	New button
- Button duplicate	Duplicate button
- Button edit	Edit button
- Button delete	Delete button
- FileChooser fileChooser	File chooser instance
- final double DIALOG_WIDTH = 300	
- final double DIALOG_HEIGHT = 600	
- final String FONT_SIZE_14 = "-fx-font-size: 14px"	

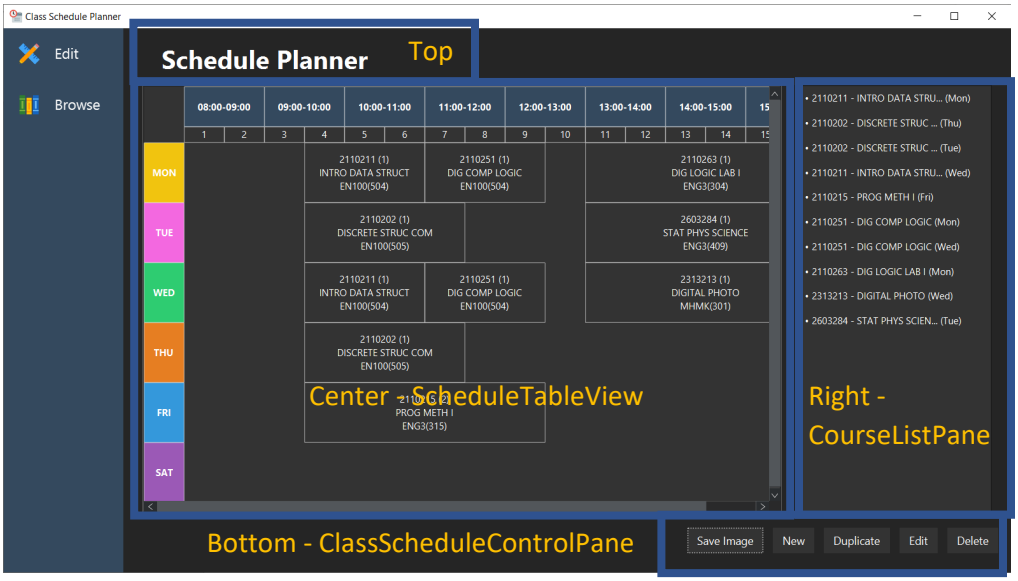
Constructor

- ClassScheduleControlPane(ClassSchedulePane schedulePane)	Initialize each component.

Method

- Stage createBaseStage()	Create stage for dialog.
- <T extends Parent> T styleRoot(T root)	Delegate call to UIUtil.styleNode()
+ void onAdd(ActionEvent event)	Called when click Add button. It will show Add dialog.
+ void onEdit(ActionEvent event)	Called when click Edit button. It will show Edit dialog.
+ void onDuplicate(ActionEvent event)	Called when click Duplicate button. It will show Add dialog with data from selected course in the list copied to the text field.
- boolean checkEligibility()	(Would never return false for current implementation) Check if currently selected course is edible.
- void showDialogUsing(Stage stage, Parent root)	Helper method to show dialog on a stage using Specified node.
+ void onSave(ActionEvent event)	Called when click Save button. It will open save file dialog to let user choose the location of file to be saved.
+ void onDelete(ActionEvent event)	Called when click Delete button. Will delete course from both the list and the table.

Class ClassSchedulePane extends BorderPane



Field

- ScheduleTableView tableView	ScheduleTableView component. Placed at the center.
- ScheduleTableModel tableModel	Data model for the schedule table
- CourseListPane courseListPane	CourseListPane component. Placed on the right
- ClassScheduleControlPane controlPane	ClassScheduleControlPane component. Placed at the bottom
- ScrollPane tableScroll	ScrollPane for table view.
- Label title	Title named “Schedule Planner”. Placed at the top.

Constructor

+ ClassSchedulePane(Arisu app)	Initialize each component then set them to their respective position. Call UIUtil.fixBlurryScrollPane(tableScroll) to make it not blurry.

Method

- Label createTitle(String name)	Helper method to create label for the title.
Getter for every fields except tableScroll and title	

Class CourseCell extends LabelCell implements ScheduleTableCell

Represent course cell on the grid of the table.

Field

- DayOfWeek day	Day
- LocalTimeRange timeRange	Time range
- String courseNumber	Course number e.g. “2110215”
- String courseName	Course name e.g. “PROG METH I” (short name)
- int section	Section number
- String teachingMethod	Teaching method e.g. “LECT”, “LAB”
- String building	Building
- String room	Room
- CourseAdapter origin	Which courseAdapter has created this cell?

Constructor

+ CourseCell(DayOfWeek day, LocalTimeRange timeRange, String courseNumber, String courseName, int section, String building, String room, String teachingMethod)	Initialize each field using these parameters. Then call setText(buildText())
+ CourseCell(DayOfWeek day, LocalTimeRange timeRange, String courseNumber, String courseName, int section)	Delegate constructor with three last parameters null.

Method

- String buildText()	Generate text to be display on the table cell.
Getter for every fields and setter for origin field.	
+ String toString()	Return string representation for this object.

Class CourseListPane extends ListView<CourseAdapter>

Field

- ClassSchedulePane schedulePane	Reference to schedule pane (its parent).
- CourseAdapter selected	Selected item on the list.
- Label empty	Label to be shown when list is empty.

Constructor

+ CourseListPane(ClassSchedulePane schedulePane)	Initialize each component. - Set cell factory to this::createListCell - Add listener to selection model to track change of the selection. - Add listener to key press action

Method

- void onKeyPressed(KeyEvent e)	Delegate action to control pane - DELETE -> delete - ENTER -> edit
- ListCell<CourseAdapter> createListCell(ListView<CourseAdapter> list)	Factory method to create custom list cell.
- void changed(ObservableValue<? extends CourseAdapter> ov, CourseAdapter oldVal, CourseAdapter newVal)	Listen for change in the selection. Then set a new value to the selected field.
+ CourseAdapter getSelected()	Get currently selected CourseAdaptor object.

Class CourseListPane.CourseListCell extends ListCell<CourseAdapter>

Method

- void updateItem(CourseAdapter item, boolean empty)	Set appropriate tooltip for each list. Call item.updateListCell(this) to update data of this list entry.

Class LabelCell extends Label

Represent plain text cell on the grid of the table.

Constructor

+ LabelCell(String text)	Initialize label. Set layout parameter to make it suitable for displaying on the grid pane.
+ LabelCell(String text, double width, double height)	Delegate call to above constructor. Then set preferred size for the label cell.

Class ModifyCourseDialog extends VBox

A dialog for modification of course on the list and the schedule table using a form.

Field

# Label courseNoLabel = new Label("Course Number")	Course number label
# TextField courseNo = new TextField("")	Course number text field
# Label courseNameLabel = new Label("Course Name")	Course name label
# TextField courseName = new TextField("")	Course name text field
# Label sectionLabel = new Label("Section")	Section label
# TextField section = new TextField("1")	Section text field
# Label dayLabel = new Label("Day")	Day of week label
# ComboBox<DayOfWeek> day = new ComboBox<>(dayOfWeekList())	Day of week combo box
# Label timeLabel = new Label("Time")	Time range label
# TimePicker time = new TimePicker()	Time picker component
# Label buildingLabel = new Label("Building (Optional)")	Building label
# TextField building = new TextField("")	Building text field
# Label roomLabel = new Label("Room (Optional)")	Room label
# TextField room = new TextField("")	Room text field
# Button commitChange = new Button("Add")	Add/Edit button
# Button cancel = new Button("Cancel")	Cancel button

# HBox buttonContainer = new HBox()	Container for button
# boolean editMode = false;	Is this dialog on the edit mode?
- SingleCourseEntry currentCourseEntry	Current course entry for edits mode. Use as an exception when check for time conflict.
# Stage stage	Current stage for this dialog
# ClassSchedulePane schedulePane	Reference to schedule pane (its parent)
- final String FONT_SIZE_14 = "-fx-font-size: 14px"	Font size css string

Constructor

+ ModifyCourseDialog(Stage stage, ClassSchedulePane schedulePane)	Initialize each component. Set listener for all buttons. Add to the children.

Method

+ void setEditMode(boolean edit)	Set editMode boolean value. - If editMode is true -> commitChange.setText("Save") - else -> commitChange.setText("Edit")
+ void feedCurrentData(SingleCourseEntry currentCourseEntry)	- Assign parameter to currentCourseEntry field. - call copyDataFrom(currentCourseEntry)
+ void copyDataFrom(SingleCourseEntry courseEntry)	Read data from SingleCourseEntry then set it back to their respective text field.
- void validateInput()	Validate input. If there are any errors then it will raise an exception.
+ void onChangeCommit(ActionEvent event)	Attempt to add or edit course. - If there aren't any problems, then it will save or edit. - Else there will have an error dialog shown to the user.
+ void onCancel(ActionEvent event)	Close current dialog(stage).
- ObservableList<DayOfWeek> dayOfWeekList()	Generate list of day in the week except Sunday.
`- ListCell<DayOfWeek> makeDayCell(ListView<DayOfWeek> param)`	Delegate call to makeDayCell()
- ListCell<DayOfWeek> makeDayCell()	Return new anonymous object that extends ListCell<DayOfWeek>(). When updateItem is called, it will set text to string

Interface ScheduleTableCell

Represent table cell, that is it has a coordinate (day and time range) associate with it.

Method

+ DayOfWeek getDay()	Get day.
+ LocalTimeRange getTimeRange()	Get time range.

Class ScheduleTableView extends GridPane

Field

final double DEFAULT_CELL_WIDTH = 100	
final double DEFAULT_CELL_HEIGHT = 75	
final LocalTime TIME_LOWER_LIMIT = LocalTime.of(8, 0)	Minimum time on the table
final LocalTime TIME_UPPER_LIMIT = LocalTime.of(16, 0);	Maximum time on the table

Constructor

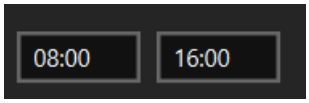
- ScheduleTableView(ClassSchedulePane schedulePane)	Set background of the grid pane. Then call initDayOfWeek() and initTime() to create table header and side header.

Method

+ void addCell(ScheduleTableCell cell)	Add cell to the table.
+ void addAllCell(ScheduleTableCell... cells)	Variadic arguments version of addCell().
+ void removeCell(ScheduleTableCell cell)	Add ScheduleTableCell type of node to the table.
+ void addNodeAt(Node node, DayOfWeek day, LocalTimeRange range)	Add node manually using day and time range.
+ int removeNodeAt(DayOfWeek day, LocalTimeRange range)	Remove cell using day and time range.
- void validateArgs(DayOfWeek day, LocalTimeRange range)	Validate if day and time range is supported by this table.
- int calculateRow(DayOfWeek day, LocalTimeRange range)	Calculate row position.
- int calculateCol(DayOfWeek day, LocalTimeRange range)	Calculate column position.
- int calculateColSpan(DayOfWeek day, LocalTimeRange range)	Calculate how many column should cell span for specified time range.
- void initDayOfWeek()	Create day row.
- void initTime()	Create time column.

Class TimePicker extends HBox

Time input field with a formatter.



Field

- TextField start	Start time text field
- TextField end	End time text field
- final SimpleDateFormat format = new SimpleDateFormat("HH:mm")	Formatter for text field

Constructor

+ TimePicker()	Initialize each component, set text formatter of start and end to format (SimpleDateFormat).

Method

+ LocalTimeRange getTimeRange()	Get LocalTimeRange object from user input. If it's invalid, it will throw an exception.
+ TextField getStart()	Get start text field.
+ TextField getEnd()	Get end text field.

Package arisu.util

A collection of utility class.

Enum DayOfWeekColor

Color representation for each day in the week.

Enum value

+ MONDAY	Store the color of Monday.
+ TUESDAY	Store the color of Tuesday.
+ WEDNESDAY	Store the color of Wednesday.
+ THURSDAY	Store the color of Thursday.

+ FRIDAY	Store the color of Friday.
+ SATURDAY	Store the color of Saturday.
+ SUNDAY	Store the color of Sunday.

Field

Color color	Color of the day.

Constructor

- DayOfWeekColor(Color color)	Create enum with specified color.

Method

+ Color getColor()	Return the color representation for that day.
DayOfWeekColor of(DayOfWeek day)	Static helper method to get color of some day.

Class LocalTimeRange

Represent a time range using start LocalTime and end LocalTime

Field

- LocalTime start	Start of time range
- LocalTime end	End of time range

Constructor

+ LocalTimeRange(LocalTime start, LocalTime end)	Construct time range from two LocalTime object, start and end.
+ LocalTimeRange(LocalTimeRange range)	Copy constructor.
+ LocalTimeRange(LocalTime start, Duration duration)	Construct time range from time point and duration of that time range.
+ LocalTimeRange(int hours1, int minutes1, int hours2, int minutes2)	Helper constructor of LocalTimeRange(start, end).

Method

+ LocalTime start()	Get the start time.
+ LocalTime end()	Get the end time.
+ long duration(TemporalUnit unit)	Get the duration between the start and the end.
+ isValidRange()	Return whether the start time comes before or equals the end time.
+ overlapWith(LocalTimeRange another)	Check if the time range of this overlaps with another one or not.

+ int hashCode()	Generate hash code.
+ boolean equals(Object obj)	Check whether this time range is equals to another one?
+ String toString()	Generate string representation which is in form of “HH:mm-HH:mm”

Class SimpleBackground

Helper class for creating a background.

Method

+ Background paint(Paint paint)	Create a background from Paint object.
+ Background rgb(int red, int green, int blue, int opacity)	Create a background from red, green, blue parameters with an opacity.
+ Background rgb(int red, int green, int blue)	Create a background from red, green, blue parameters.
+ Background web(String colorString, double opacity)	Create a background from a hex color code with an opacity.
+ Background web(String colorString)	Create a background from a hex color code.

Class SimpleBorder

Helper class for creating a border.

Method

+ solid(Paint paint, double thickness)	Create solid border with specified Paint and thickness.
+ solid(Paint paint, BorderWidths widths)	Create solid border with specified Paint and BorderWidths object.

Class UIUtil

A collection of helper method to help in construction of UI component.

Method

+ void fixBlurryScrollPane(ScrollPane ...scrollPanes)	Fix blurry element inside the ScrollPane by disabling cache of StackPane inside a ScrollPane. (Workaround for JavaFX bug)

+ void styleNode(Parent parent)	Add JMetro style to the specified parent Node. Use when creating a new stage.
+ Alert makeAlert(AlertType alertType, String contentText, ButtonType... buttons)	Make an alert with customized style.
+ Alert makeAlert(AlertType alertType)	Delegate to another makeAlert method.
+ useSizeAdapter(Scene scene, Region node)	Add a listener to node to make it change size according to size of scene.