

# CRC CARDS – HAUNTED PYRAMID

[HTTPS://GITHUB.COM/CSC207-UOFT/COURSE-PROJECT-HAUNTED-PYRAMID](https://github.com/CSC207-UOFT/COURSE-PROJECT-HAUNTED-PYRAMID)

OCTOBER 15<sup>TH</sup>, 2021

# SUMMARY

- Entities
  - User
  - Student
  - Event
  - Course
  - OurCalendar
- To be implemented
  - Lecture
  - Reading
  - Test
  - Tutorial
  - Labs
  - Others
- Use Cases
  - CalendarManager
  - CourseManager
  - EventManager
  - StudentManager
- Controllers
  - CalendarController
  - CourseController
  - EventController
  - LoginController
  - MainController
  - StudentController
- Interface
  - Fluid
  - Repeated
- Presenter
  - CalendarPresenter
- Gateway
  - IOSerializable

# ENTITY User

Student

Responsibilities	Collaborators
<ul style="list-style-type: none"><li>• Id</li><li>• Name</li><li>• Username</li><li>• Password</li><li>• birthDate</li><li>• dateCreated</li><li>• emailAddress</li><li>• phoneNumber</li><li>• homeAddress</li></ul>	
<ul style="list-style-type: none"><li>• Setter of the above responsibilities</li></ul>	
<ul style="list-style-type: none"><li>• Getter of the above responsibilities</li></ul>	
<ul style="list-style-type: none"><li>• Construct with name and birthDate</li></ul>	

ENTITY

# Student

Responsibilities	Collaborators
<ul style="list-style-type: none"> <li>• number of courses</li> <li>• courses enrolled</li> </ul>	<ul style="list-style-type: none"> <li>• Course</li> </ul>
<ul style="list-style-type: none"> <li>• AddCourse</li> <li>• RemoveCourse</li> <li>• Getter methods</li> </ul>	

# ABSTRACT

## ENTITY Event

Reading, Assignment, Lecture,  
Lab, Tutorial, Test, Other

Responsibilities	Collaborators
<ul style="list-style-type: none"><li>• name</li><li>• location (Optional)</li><li>• startTime</li><li>• endTime</li></ul>	
<ul style="list-style-type: none"><li>• modify startTime and endTime</li></ul>	
<ul style="list-style-type: none"><li>• Setters for startTime, endTime, name and location</li></ul>	
<ul style="list-style-type: none"><li>• Getters for startTime, endTime and length</li></ul>	
<ul style="list-style-type: none"><li>• HasConflict</li></ul>	
<ul style="list-style-type: none"><li>• Equals</li></ul>	
<ul style="list-style-type: none"><li>• ToString</li></ul>	

## ENTITY Assignment

Responsibilities	Collaborators
<ul style="list-style-type: none"><li>• worksessionsLength</li></ul>	<ul style="list-style-type: none"><li>• Fluid</li></ul>
<ul style="list-style-type: none"><li>• hoursSpent</li></ul>	
<ul style="list-style-type: none"><li>• hoursNeeded</li></ul>	
<ul style="list-style-type: none"><li>• DueDate</li></ul>	

# ENTITY

## Course

Responsibilities	Collaborators
<ul style="list-style-type: none"><li>• name</li><li>• EventList [events share the same course attribute organized in a map based on event type]</li></ul>	<ul style="list-style-type: none"><li>• Event</li><li>• Lecture</li><li>• Assignment</li><li>• Reading</li><li>• Test</li><li>• Tutorial</li><li>• Labs</li><li>• Others</li></ul>
<ul style="list-style-type: none"><li>• Add events to task</li></ul>	
<ul style="list-style-type: none"><li>• Remove events from task</li></ul>	
<ul style="list-style-type: none"><li>• Getter of name and task</li></ul>	

## ENTITY

# OurCalendar

Responsibilities	Collaborators
<ul style="list-style-type: none"><li>• calendarMap</li><li>• conflict (Boolean)</li><li>• conflictEvent</li><li>• dateInfo</li><li>• calendarMap</li></ul>	<ul style="list-style-type: none"><li>• Event</li><li>• Lecture</li><li>• Assignment</li><li>• Reading</li><li>• Test</li><li>• Tutorial</li><li>• Labs</li><li>• Others</li></ul>
<ul style="list-style-type: none"><li>• add and remove objects from calendarMap</li></ul>	
<ul style="list-style-type: none"><li>• check conflicts on calendarMap</li></ul>	
<ul style="list-style-type: none"><li>• Getter of conflictEvent, conflict, dateInfo and calendarMap</li></ul>	



# USE CASE CalendarManager

Responsibilities	Collaborators
<ul style="list-style-type: none"> <li>• CurrentManager</li> <li>• futureManager</li> <li>• pastCalendar</li> <li>• currentYear, currentMonth, currentDate</li> </ul>	<ul style="list-style-type: none"> <li>• OurCalendar</li> <li>• Lecture</li> <li>• Assignment</li> <li>• Reading</li> <li>• Test</li> <li>• Tutorial</li> <li>• Labs</li> <li>• Others</li> <li>• Event</li> </ul>
<ul style="list-style-type: none"> <li>• Generate and store past and future calendars</li> </ul>	
<ul style="list-style-type: none"> <li>• Getters for Monthly Calendar, Weekly Calendar and Yearly Calendar, currentYear, currentMonth, currentDate</li> </ul>	
<ul style="list-style-type: none"> <li>• Add and remove events to/from a Calendar</li> </ul>	
<ul style="list-style-type: none"> <li>• Notify conflict events for chosen date</li> <li>• Get event time and date info for the selected date</li> </ul>	

## USE CASE EventManager

Responsibilities	Collaborators
<ul style="list-style-type: none"><li>• Map of Events</li><li>• Map of Unscheduled Hours</li><li>• Map of Scheduled Hours</li><li>• getDay</li><li>• getType</li><li>• GetEvent</li><li>• RemoveEvent</li><li>• Rescheduable</li><li>• addEvent</li><li>• replaceEvent</li><li>• totalHours</li><li>• freeTime</li><li>• chronological</li></ul>	<ul style="list-style-type: none"><li>• CalendarManager</li><li>• Event</li><li>• Fluid</li><li>• Repeated</li></ul>

## USE CASE

# StudentManager

Responsibilities	Collaborators
<ul style="list-style-type: none"><li>• Student id to student courses map</li><li>• Student id to student object map</li><li>• Create a studentMap with given information and acquired courseList from CourseManager.</li><li>• Store student information in studentInfo</li><li>• Check username and password</li><li>• getStudentInfo</li><li>• getStudentMap</li><li>• addNewStudent</li></ul>	<ul style="list-style-type: none"><li>• Student</li><li>• Course</li></ul>

## USE CASE CourseManager

Responsibilities	Collaborators
<ul style="list-style-type: none"><li>• courseList (array of Course's that the events created belong to)</li><li>• Create courseList of the current user.</li><li>• Getter of courseList</li><li>• Getter of one specific course information from courseList</li><li>• Add Course to courseList</li><li>• Remove Course from courseList</li></ul>	<ul style="list-style-type: none"><li>• Course</li><li>• Event</li><li>• Student</li></ul>

## CONTROLLER

# MainController

Responsibilities	Collaborators
<ul style="list-style-type: none"><li>Instantiates IOSerializable</li><li>Instantiates StudentController and LoginController and initially checks the verification of the user through log in phase</li><li>Instantiates EventController</li><li>Instantiates CalendarPresenter</li><li>DisplayInitScreen</li><li>DisplayCalendar</li><li>generateEvent</li></ul>	<ul style="list-style-type: none"><li>*CalendarManager</li><li>*EventManager</li><li>LoginController</li><li>StudentController</li><li>EventController</li><li>CalendarPresenter</li><li>IOSerializable</li></ul>
<ul style="list-style-type: none"><li>saveAndExitProgram</li></ul>	<p>*Managers will move into their own controllers after phase 0 for better organization and simplicity in the MainController</p>

## CONTROLLER

# LoginController

Responsibilities	Collaborators
<ul style="list-style-type: none"><li>• login</li><li>• logout</li><li>• signup</li><li>• isLoggedIn</li></ul>	<ul style="list-style-type: none"><li>• StudentController</li></ul>

## CONTROLLERS

# EventController

Responsibilities	Collaborators
Instantiate EventManager and CalendarManager	<ul style="list-style-type: none"><li>• CalendarManager</li><li>• EventManager</li><li>• CalendarPresenter</li></ul>
Schedule – take user input to schedule, modify, remove events	

## CONTROLLERS

# StudentController

Responsibilities	Collaborators
<ul style="list-style-type: none"><li>Initialize StudentManager based on readability of the serializable file. If there isn't a saved data in the directory, create a sample file: "students.ser" with sample login information. If there is a file, read from it and instantiate StudentManager with the arraylist of students obtained from the serializable file.</li></ul>	<ul style="list-style-type: none"><li>IOSerializable</li><li>StudentManager</li></ul>



Fluid

Responsibilities	Collaborators
<ul style="list-style-type: none"><li>• Setter for fluid hours remaining</li></ul>	<ul style="list-style-type: none"><li>• AutoSchedule</li></ul>
<ul style="list-style-type: none"><li>• Getter for fluid hours remaining</li></ul>	
<ul style="list-style-type: none"><li>• Setter for work session length</li></ul>	
<ul style="list-style-type: none"><li>• Getter for work session length</li></ul>	
<ul style="list-style-type: none"><li>• Getter for session</li></ul>	
<ul style="list-style-type: none"><li>• scheduleSession</li></ul>	

# AutoSchedule

Responsibilities	Collaborators
<ul style="list-style-type: none"><li>• Setter for fluid hours remaining</li></ul>	

Repeated

Responsibilities	Collaborators
<ul style="list-style-type: none"><li>• Setter for repitition</li></ul>	
<ul style="list-style-type: none"><li>• Getter for list and start times</li></ul>	
<ul style="list-style-type: none"><li>• @Override conflicts</li></ul>	

# IOSerializable

Responsibilities	Collaborators
<ul style="list-style-type: none"><li>• Logger – logs any errors or exceptions</li><li>• EVENTS_FILEPATH</li><li>• TASKS_FILEPATH</li><li>• STUDENTS_FILEPATH</li><li>• CALENDAR_FILEPATH</li><li>• hasSavedData - checks if there is any saved data in the directory and return a boolean</li></ul>	<p><b>**Student</b> – this is necessary because we need to type cast the read object as an <code>ArrayList&lt;Student&gt;</code> when deserializing the file. This could change through code optimization or new insight for later phases of the program.</p>
<ul style="list-style-type: none"><li>• studentsReadFromSerializable – read from "students.ser" and return an ArrayList of Students that were saved in the directory</li><li>• studentsWriteToSerializable – write to "students.ser" and save the ArrayList of students</li></ul>	
<ul style="list-style-type: none"><li>• <b>***</b>It is important to note that for phase 0, serialization of students were only implemented fully. Going forward, most if not all entities will follow</li></ul>	

PRESENTER

CalendarPresenter

Responsibilities	Collaborators
<ul style="list-style-type: none"><li>Starts from an empty calendar</li></ul>	<ul style="list-style-type: none"><li>Calendar Manager</li></ul>
<ul style="list-style-type: none"><li>Add information from the calendar manager</li></ul>	
<ul style="list-style-type: none"><li>Display information as string from calendar manager</li></ul>	