

# Module Delivery Plans: Assessment Data Collection and Visualization

---

The OASIS (Team 6)

# Our Team

Ahmad Bilal

Chang Luo

Josh Hare (Software Team Leader)

Emily Lau (Project Manager)

Hongyu Tang

Ting Zhang

# The Current System

- Learning and Teaching Manager sends around Google form for lecturers to fill out module information
  - Tutors manually check assessment dates for bunching
  - Tutors manually convert data into module booklet
  - User-unfriendly!
-

# The Project Objectives

Create a user-friendly alternative to the current system

- Create a way for module leaders to enter data themselves
  - Be able to validate module information automatically
  - Visualize assessment occurrences throughout a semester
-

# **Sprint 1**

Planning, setting up  
front end and back  
end

# Sprint 1 User Stories

- As an academic tutor, I want to check if the assessments in a module add up to 100%, so that the students grades are calculated correctly
- As an academic tutor, I want to check the assessment dates are within the semester set for the module, so that students are able to complete assessments on time
- As an academic tutor, I want to check the assessment dates are not on weekends and public holidays, so that students can maintain a good work-life balance

# Our division of labor

## **Front end:**

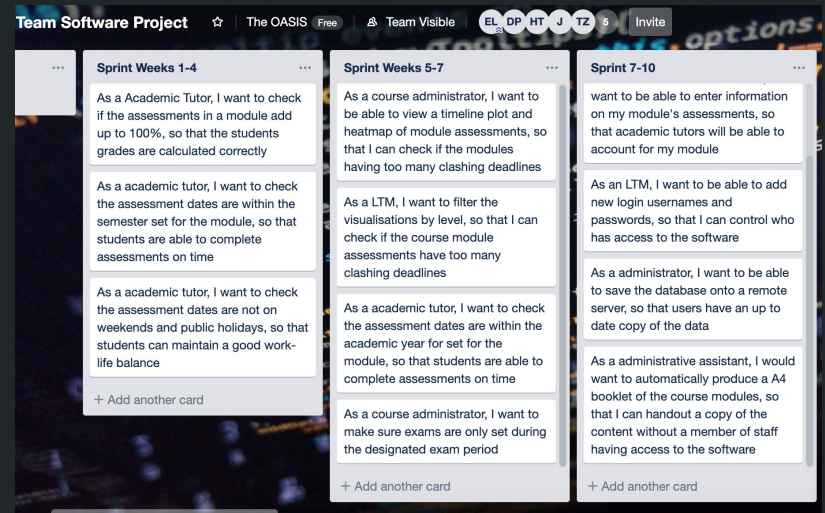
- Chang Luo
- Hongyu Tang
- Ting Zhang

## **Back end:**

- Ahmad Bilal
- Josh Hare
- Emily Lau

# Our organization tools

- Met once a week for three hours
- Facebook group chat to organize meetings and communicate about work outside meeting times
- Trello to organize task division
- Git Lab repository for storing team code





# Back End activities

- Divide data set into “module” and “assessment” classes
- Wrote functions that validate the assessment data for each module

```
class module(object):
    def __init__(self, module_name, module_teacher, credit, assessment_list, semester, timestamp):
        ''' input modules_name = str, module_teacher = str, assessment_list = [class(assessment)],
            semester = str(1,2,BOTH), semester_date_range = [semester_start_date, semester_end_date]
            timestamp = str(time last edited)
        '''
        self.module_name = module_name
        self.module_teacher = module_teacher
        self.credit = credit
        self.assessment_list = assessment_list
        self.semester = semester
        self.timestamp = timestamp
```

```
class assessment(object):
    def __init__(self, handout_date, handin_date, assessment_type, assessment_name, percentage):
        self.handout_date = handout_date #str
        self.handin_date = handin_date #str
        self.assessment_type = assessment_type #str
        self.assessment_name = assessment_name #str
        self.percentage = percentage #float

    def __str__(self):
        STR = ''.join("hand-out date " + str(self.handout_date))
        STR +=("\nhand-in date " + str(self.handin_date))
        STR +=("\nassessment type " + str(self.assessment_type))
        STR +=("\nassessment name " + str(self.assessment_name))
        STR +=("\npercentage of module " + str(self.percentage))
        return STR

    def check_percentage(self):
        sum_perc = 0
        for ass in self.assessment_list:
            sum_perc += float(ass.percentage)
        if(sum_perc != 100):
            err = "\nERROR, sum of module assessments not equal to 100 in: " + str(self.module_name)
            return err
            #maybe we should send a message to the interface?
        else:
            return ''

    def check_dates(self):
        for assessment in self.assessment_list:
            raiseFlag = False
            if(checkdate.check_date(assessment) == False):
                raiseFlag = True
            if(checkdate.academic_year(assessment) == False):
                raiseFlag = True
            if(checkdate.check_weekday(assessment) == False):
                raiseFlag = True

        if(raiseFlag == True):
            # Todo create popup in GUI
            err = str(self.module_name) + "\n" + assessment.__str__() + "\n" ##continue error print from check_date()
            return err
        else:
            return ''
```

# Front End activities

- Designed the interface according to requirements
- Created the interface by PyQt5
- Used QSS to stylize the interface
- Some widgets used frequently are written as class

```
QPushButton#function_button{
    border:none;
    color:white;
    font-size:20px;
    font-weight:400;
    text-align:center;
}
QPushButton#function_button:hover{
    color:#FFFAFA;
    font-weight:600;
}
QWidget{
    border-left:0px;
    background:#3367D6;
}
```

```
class Header(QWidget):
    def __init__(self, parent=None):
        super(Header, self).__init__(parent)
        self.flag = False
        self.setObjectName('Header')

        self.parent = parent

        self.header_layout = QHBoxLayout()
        self.setLayout(self.header_layout)

        # create button for header
        self.left_close = QtWidgets.QPushButton(html.unescape('&times;'))
        self.left_close.setFixedSize(16, 16)
        self.left_close.clicked.connect(QCoreApplication.quit)
        with open('QSS/left_close.qss', 'r') as f:
            style = f.read()
            self.left_close.setStyleSheet(style)
```

# Sprint 1 User Stories - completed?

- As an academic tutor, I want to check if the assessments in a module add up to 100%, so that the students grades are calculated correctly ✓
- As an academic tutor, I want to check the assessment dates are within the semester set for the module, so that students are able to complete assessments on time ✓
- As an academic tutor, I want to check the assessment dates are not on weekends and public holidays, so that students can maintain a good work-life balance ✓

# **Sprint 2**

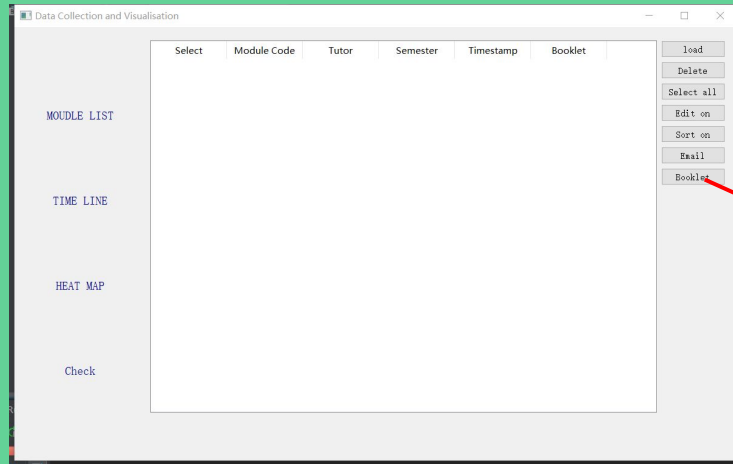
Bringing front and  
back end together

# Sprint 2 User Stories

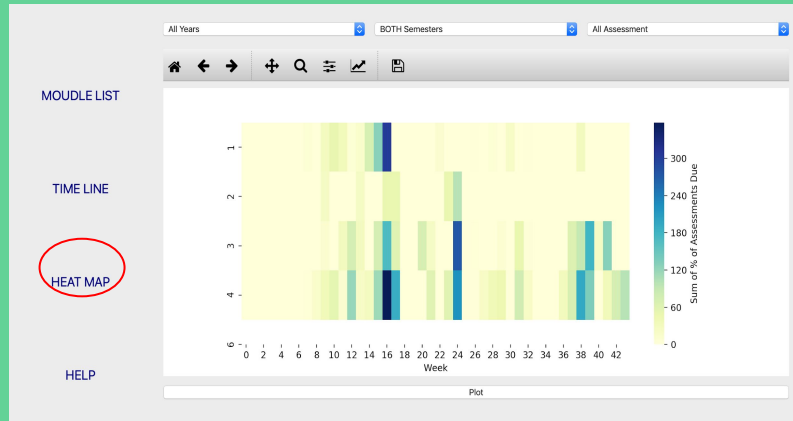
- As a course administrator, I want to be able to view a timeline plot and heatmap of module assessments, so that I can check if the modules having too many clashing deadlines
- As a LTM, I want to filter the visualisations by level, so that I can check if the course module assessments have too many clashing deadlines
- As a academic tutor, I want to check the assessment dates are within the academic year for set for the module, so that students are able to complete assessments on time
- As a course administrator, I want to make sure exams are only set during the designated exam period

# Interface Prototypes

Interface  
Prototype



Heatmap  
sample



A screenshot of a form titled "My Booklet". The form contains the following fields and controls:

- Module xxx**
- Academic Leader :** name
- Semester :** Semester 1 (or 2) only
- Assessment :**

Assessment	Date	Worth
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

At the bottom of the form is a "Print" button.

Early version of  
module booklet

# Some challenges with this sprint

- Joining up front end and back end - separate bodies of code
  - Some tasks overlapped heavily - some people accidentally did other people's tasks
  - Learning how PyQt5 works, and how to embed visuals
-

# Sprint 2 User Stories - completed?

- As a course administrator, I want to be able to view a timeline plot and heatmap of module assessments, so that I can check if the modules having too many clashing deadlines ✓
- As a LTM, I want to filter the visualisations by level, so that I can check if the course module assessments have too many clashing deadlines ✓
- As a academic tutor, I want to check the assessment dates are within the academic year for set for the module, so that students are able to complete assessments on time ✓
- As a course administrator, I want to make sure exams are only set during the designated exam period ✓



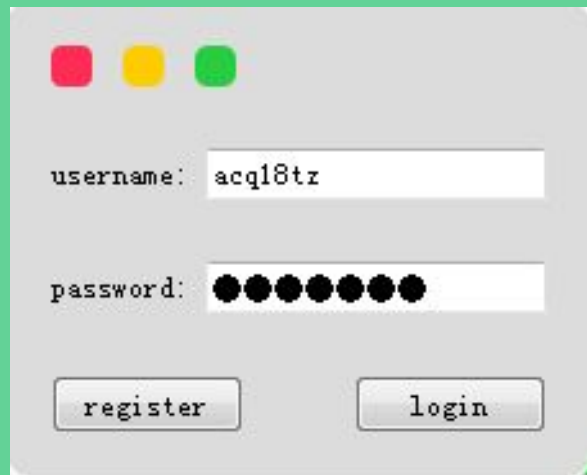
# **Sprint 3**

Refining the interface

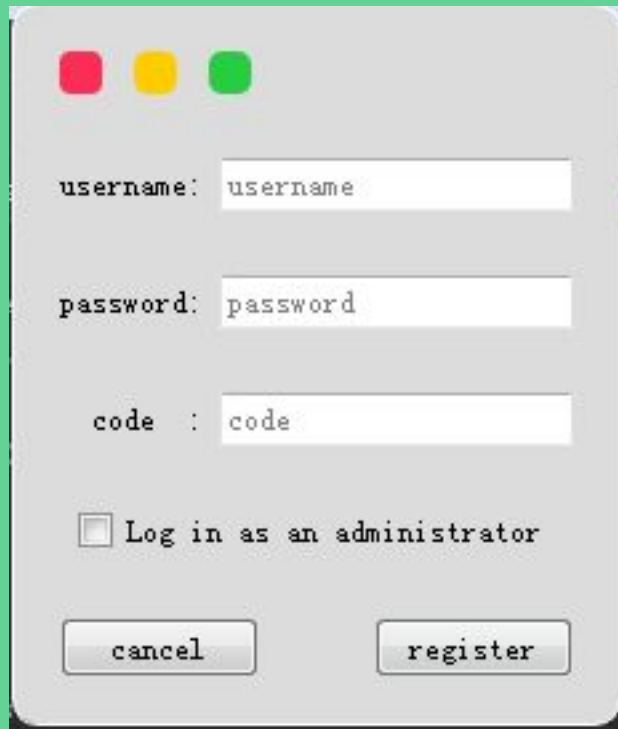
# Sprint 3 User Stories

- As a user, I want a login and password protection, so that only verified colleagues can alter the data
- As a member of academic staff, I want to be able to enter information on my module's assessments, so that academic tutors will be able to account for my module
- As an LTM, I want to be able to add new login usernames and passwords, so that I can control who has access to the software
- As an LTM, I want to be able to save the database onto a remote server, so that users have an up to date copy of the data
- As an LTM, I would want to automatically produce a A4 booklet of the course modules, so that I can handout a copy of the content without a member of staff having access to the software

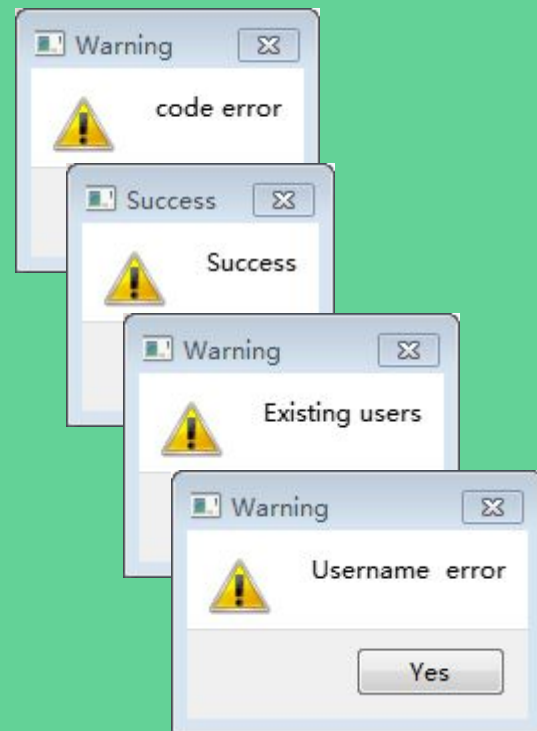
# Final Product



A login form window with a title bar containing three colored buttons (red, yellow, green). The form contains two input fields: "username:" with the text "acq18tz" and "password:" with ten black dots. Below the fields are two buttons: "register" and "login".



A registration form window with a title bar containing three colored buttons (red, yellow, green). The form contains three input fields: "username:" with the text "username", "password:" with the text "password", and "code :" with the text "code". Below the fields is a checkbox labeled "Log in as an administrator" which is unchecked. At the bottom are two buttons: "cancel" and "register".



Click to load data from CSV

# Module list

The screenshot shows a Moodle interface with a sidebar on the left and a main content area. The sidebar contains navigation options: MOUDLE LIST (with a home icon), TIME LINE (with a bar chart icon), HEAT MAP (with a location pin icon), and Check (with a checkmark icon). The main content area features a toolbar with icons for folder, save, delete, refresh, copy, print, and sort. Below the toolbar is a table of modules with columns for Select, Module Code, Tutor, Semester, Timestamp, and Credits. Red arrows point from yellow callout boxes to specific elements: 'Click to load data from CSV' points to the folder icon; 'Sort modules' points to the sort icon; 'Delete module' points to the delete icon; 'Edit module information' points to the refresh icon; 'View booklet of a selected module' points to the copy icon; 'Save changes made to module info' points to the save icon; 'Select all modules' points to the select column header; and 'Delete module' also points to the delete icon. The table contains 18 rows of module data.

	Select	Module Code	Tutor	Semester	Timestamp	Credits
1	<input type="checkbox"/>	COM6102	Maria-Cruz Villa-Uriol	Semester 2 only	09/12/2018 15:08	15
2	<input type="checkbox"/>	COM1006	Ramsay Ta		9/14/2018 12:32:24	15
3	<input type="checkbox"/>	COM4509 / 6509	Haiping Lu		09/10/2018 17:37	15
4	<input type="checkbox"/>	COM4502 / 6502	Roger Moore	Semester 1 only	9/13/2018 10:57:06	15
5	<input type="checkbox"/>	COM4503 / 6503	Yoshi Gotoh	Semester 1 only	09/10/2018 12:44	15
6	<input type="checkbox"/>	COM1008	Haiping Lu	Semester 1 only	09/04/2018 11:11	15
7	<input type="checkbox"/>	COM6012	Haiping Lu (jointly wi...	Semester 2 only	09/10/2018 17:55	15
8	<input type="checkbox"/>	COM6012	Yoshi Gotoh	Semester 1 only	8/25/2018 12:57:09	15
9	<input type="checkbox"/>	COM1008	Roger Moore	Semester 1 only	9/13/2018 10:57:14	15
10	<input type="checkbox"/>	COM6012	Tony Prescott	BOTH Semesters	9/20/2018 21:43:08	15
11	<input type="checkbox"/>	COM63001	Dawn Walker	Semester 2 only	9/24/2018 13:47:46	15
12	<input type="checkbox"/>	COM6535	Harriet Holman	Semester 2 only	9/14/2018 16:50:01	15
13	<input type="checkbox"/>	COM6104	Emma Norling	Semester 2 only	09/04/2018 11:30	15
14	<input type="checkbox"/>	COM6655	Amanda Sharkey	Semester 1 only	09/12/2018 16:38	15
15	<input type="checkbox"/>	COM160/161	Mark Hepple	Semester 1 only	9/18/2018 13:48:46	15
16	<input type="checkbox"/>	COM6016	John A Clark	Semester 1 only	9/13/2018 12:15:25	15
17	<input type="checkbox"/>	COM3310	Richard Clayton	Semester 1 only	8/24/2018 15:05:07	15
18	<input type="checkbox"/>	COM2107	Georg Struth	Semester 2 only	8/28/2018 11:43:59	15

Load the data from database.

# Check

**MOUDLE LIST**

**TIME LINE**

**HEAT MAP**

**Check**

Error information: Update

COM1001  
Sum of module assessments not equal to 100

COM1002  
Handin date is not within academic year

COM1003  
Sum of module assessments not equal to 100

COM1005  
Sum of module assessments not equal to 100

COM1006  
Sum of module assessments not equal to 100

COM1009  
Handin date is not within academic year  
Handin date is on a weekend

Email:

Update the error information!

Click to update the error information

The error information of all modules

Write down your email address here and send these information to your account

# Timeline

The screenshot shows the Moodle Timeline interface. On the left is a sidebar with navigation icons for MOUDLE LIST, TIME LINE, HEAT MAP, and Check. The top of the main area has a filter bar with dropdown menus for '1st Year', 'Semester 2 only', and 'All Assessment'. Below the filter bar is a toolbar with navigation icons. The main area displays a horizontal timeline with several event markers. Each marker is labeled with the assessment type and date. A 'Display' button is located at the bottom of the timeline.

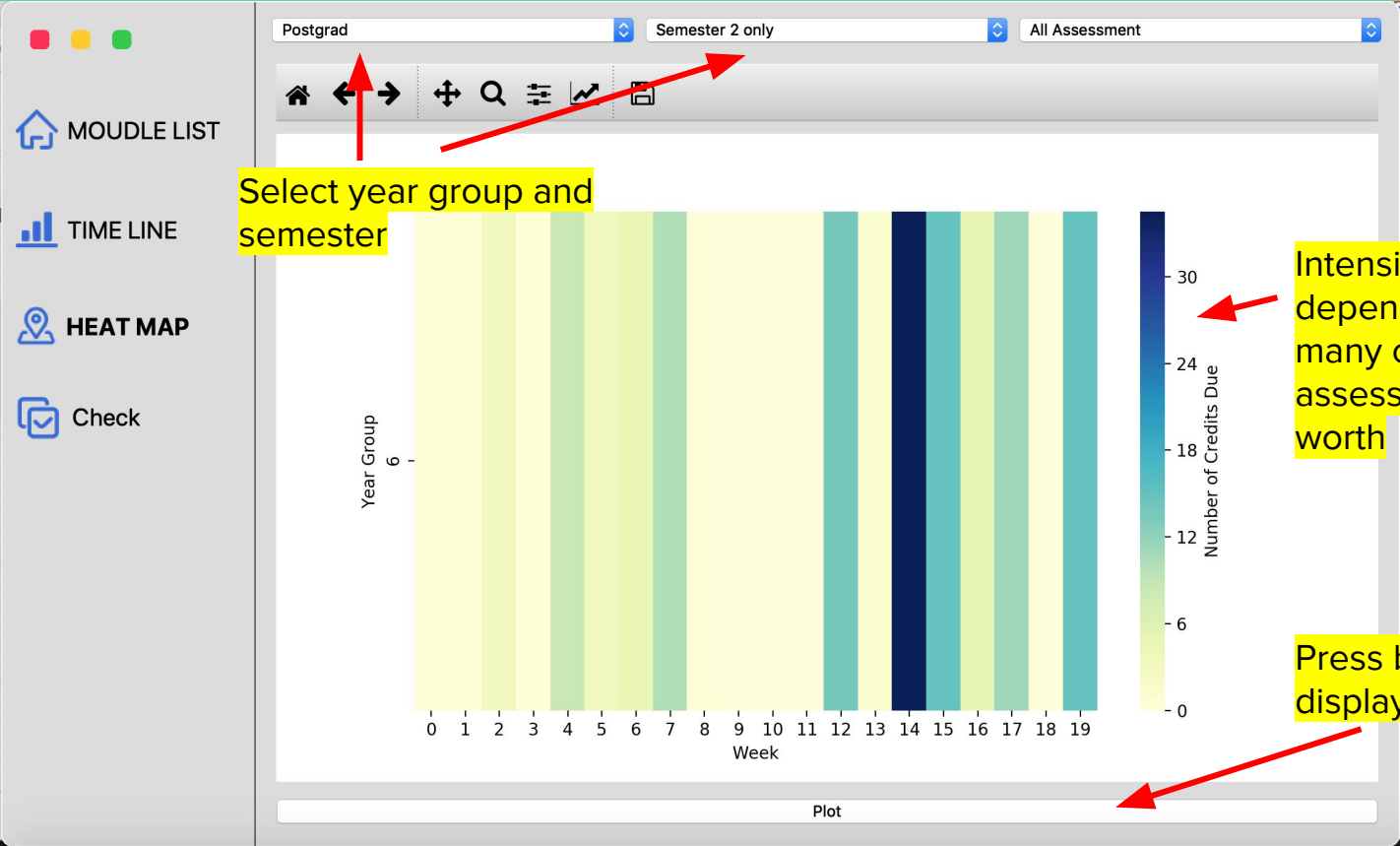
**Select year group and semester**

**Each event displays assessment type and handin date**

**Press button to display timeline**

Date	Assessment Type
6/02/2019	MOLE quiz
01/03/2019	MOLE quiz
01/03/2019	Exercise/problem sheet
15/03/2019	Exercise/problem sheet
15/03/2019	Assignment
29/03/2019	MOLE quiz
18/05/2019	Assignment
18/05/2019	MOLE quiz

# Heatmap



# Final Booklet Format and Features

COM3310 Assessments

## Module COM3310

**Academic Leader :** Richard Clayton  
**Semester :** Semester 1 only

### Assessments

Force User to enter date value

Assessment name	Assessment type	Date	Worth
1 Assignment part 1	Assignment	19/11/2018	30.00
2 Assignment part 2	Assignment	14/12/2018	30.00
3 Exam	Formal exam	01/02/2019	40.00

Force User to enter numerical value from min 0 to max 100

What percentage of the module is this assessment worth?

5
100%
5%
5
01-Jun
35%
8%

Rob Gaizauskas

Error

**Error in Assessments**

Error must be fixed before saving

Hide Details... Close

Sum of module assessments not equal to 100  
Handin date is on a weekend



# Sprint 3 User Stories - completed?

- As a user, I want a login and password protection, so that only verified colleagues can alter the data ✓
- As a member of academic staff, I want to be able to enter information on my module's assessments, so that academic tutors will be able to account for my module ✓
- As an LTM, I want to be able to add new login usernames and passwords, so that I can control who has access to the software ✓
- As an administrator, I want to be able to save the database onto a remote server, so that users have an up to date copy of the data ✓
- As an administrative assistant, I would want to automatically produce an A4 booklet of the course modules, so that I can handout a copy of the content without a member of staff having access to the software ✓

# Conclusion

- Completed all essential features named in requirements
- Made interface that is aesthetically pleasing and easy to use
- Learned new skills
- Learned to work as a team!

# Any Questions?

Thank you for listening!

The screenshot displays a Moodle course page with a sidebar on the left and a main content area on the right. The sidebar contains four navigation options: 'MOUDLE LIST' (with a house icon), 'TIME LINE' (with a bar chart icon), 'HEAT MAP' (with a location pin icon), and 'Check' (with a checkmark icon). The main content area features a table with the following columns: 'Select', 'Module Code', 'Tutor', 'Semester', 'Timestamp', and 'Credits'. The table lists 18 modules, each with a 'Select' checkbox, a 'Module Code', a 'Tutor' name, a 'Semester' designation, a 'Timestamp', and a 'Credits' value of 15. At the bottom of the main content area, there is a button labeled 'Load the data from database.'

Select	Module Code	Tutor	Semester	Timestamp	Credits
<input type="checkbox"/>	COM6102	Maria-Cruz Villa-Uriol	Semester 2 only	09/12/2018 15:08	15
<input type="checkbox"/>	COM1006	Ramsay Taylor	Semester 1 only	9/14/2018 12:32:24	15
<input type="checkbox"/>	COM4509 / 6509	Haiping Lu (jointly wi...	Semester 1 only	09/10/2018 17:37	15
<input type="checkbox"/>	COM4502 / 6502	Roger Moore	Semester 1 only	9/13/2018 10:57:06	15
<input type="checkbox"/>	COM4503 / 6503	steve maddock	Semester 1 only	09/10/2018 12:44	15
<input type="checkbox"/>	COM1008	Emma Norling	Semester 1 only	09/04/2018 11:11	15
<input type="checkbox"/>	COM6012	Haiping Lu (jointly wi...	Semester 2 only	09/10/2018 17:55	15
<input type="checkbox"/>	COM6063	Yoshi Gotoh	Semester 1 only	8/25/2018 12:57:09	15
<input type="checkbox"/>	COM3502	Roger Moore	Semester 1 only	9/13/2018 10:57:14	15
<input type="checkbox"/>	COM3330	Tony Prescott	BOTH Semesters	9/20/2018 21:43:08	15
<input type="checkbox"/>	COM3001	Dawn Walker	Semester 2 only	9/24/2018 13:47:46	15
<input type="checkbox"/>	COM6535	Harriet Holman	Semester 2 only	9/14/2018 16:50:01	15
<input type="checkbox"/>	COM6104	Emma Norling	Semester 2 only	09/04/2018 11:30	15
<input type="checkbox"/>	COM6655	Amanda Sharkey	Semester 1 only	09/12/2018 16:38	15
<input type="checkbox"/>	COM160/161	Mark Hepple	Semester 1 only	9/18/2018 13:48:46	15
<input type="checkbox"/>	COM6016	John A Clark	Semester 1 only	9/13/2018 12:15:25	15
<input type="checkbox"/>	COM3310	Richard Clayton	Semester 1 only	8/24/2018 15:05:07	15
<input type="checkbox"/>	COM2107	Georg Struth	Semester 2 only	8/28/2018 11:43:59	15

Load the data from database.