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:

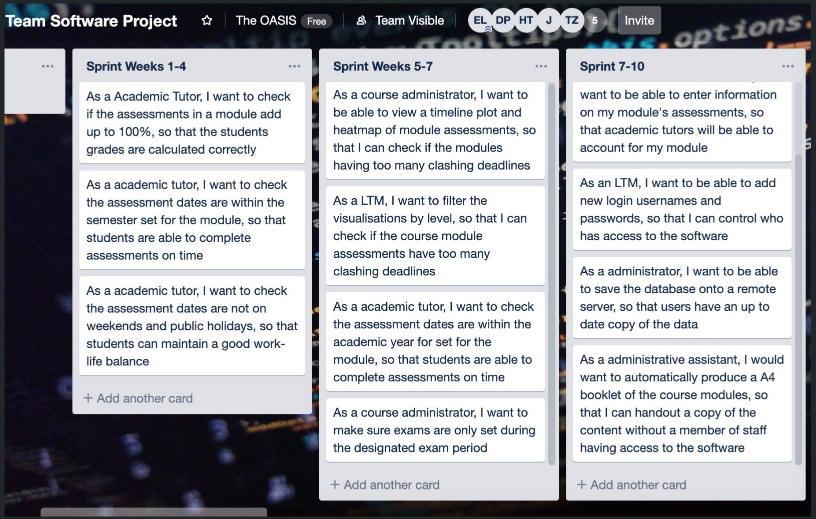
- Chang Luo
- Hongyu Tang
- Ting Zhang

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



Sprint Weeks 1-4

- As a 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 a 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 a 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

+ Add another card

Sprint Weeks 5-7

- 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

+ Add another card

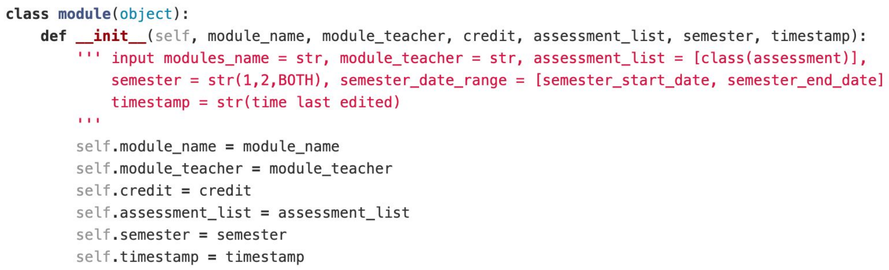
Sprint 7-10

- 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 a 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 a administrative assistant, 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

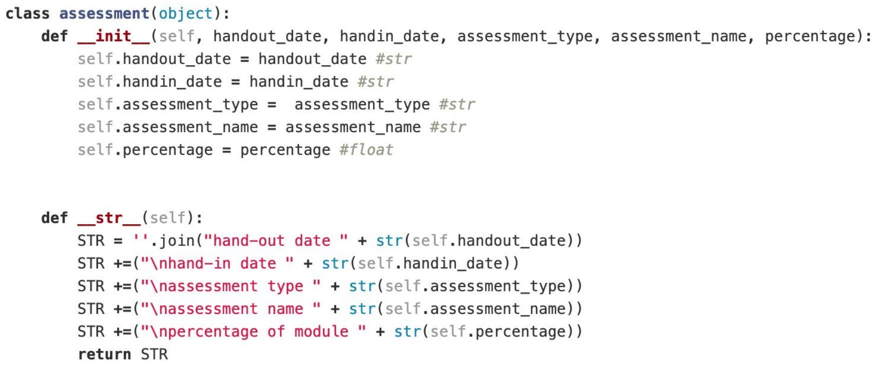
+ Add another card

Back End activities

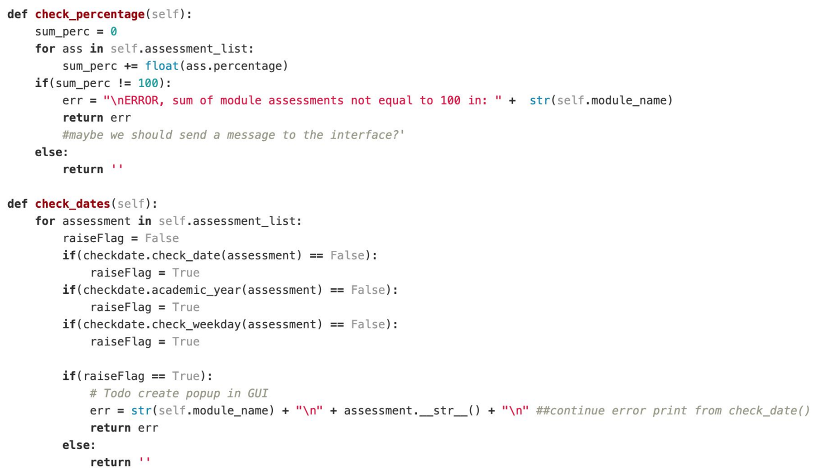
- Divide data set into “module” and “assessment” classes
- Wrote functions that validate the assessment data for each module



```
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
```



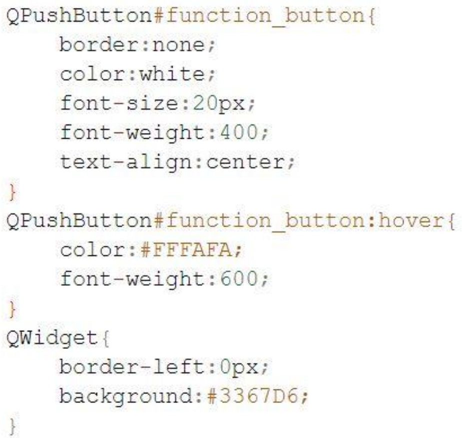
```
class assessment(object):
    def __init__(self, handout_date, handin_date, assessment_type, assessment_name, percentage):
        self.handout_date = handout_date #str
        self.handin_date = handin_date #str
        self.assessment_type =  assessment_type #str
        self.assessment_name =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



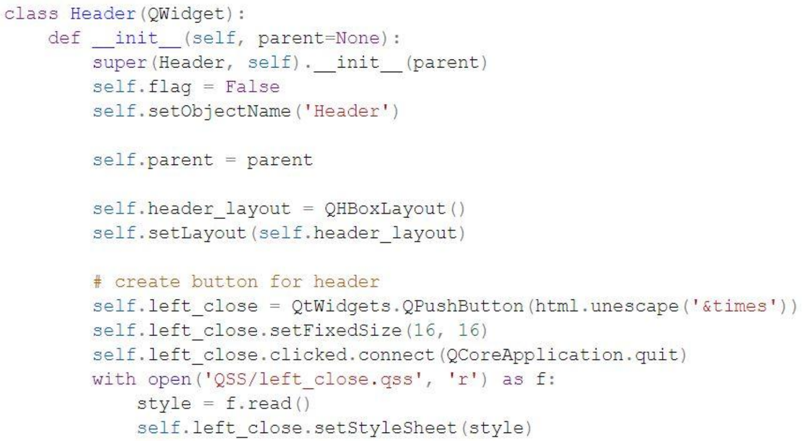
```
def check_percentage(self):
    sum_perc = 0
    for ass in self.assessment_list:
        sum_perc += float(ass.percentage)
    if(sum_perc != 100):
        err = "\nERROR, sum of module assessments not equal to 100 in: " +  str(self.module_name)
        return err
        #maybe we should send a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

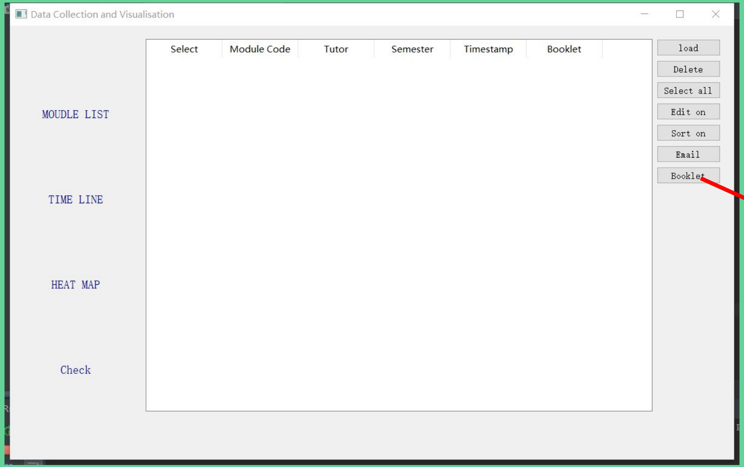
Bringing front and back end together

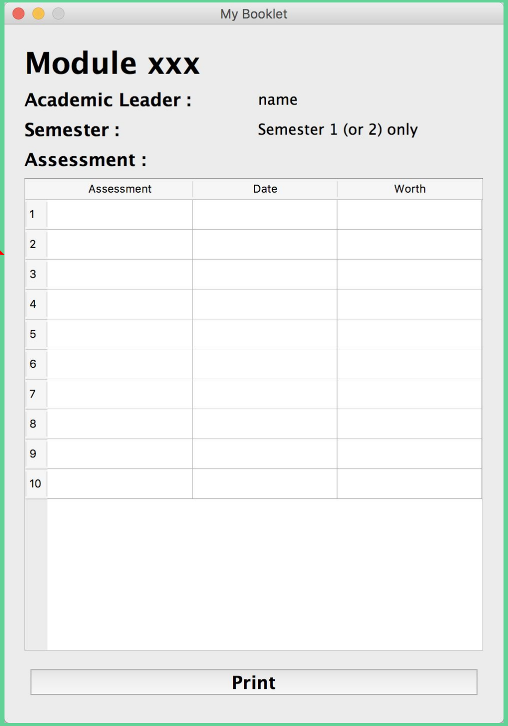
Sprint 2 User Stories

- 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





Module xxx

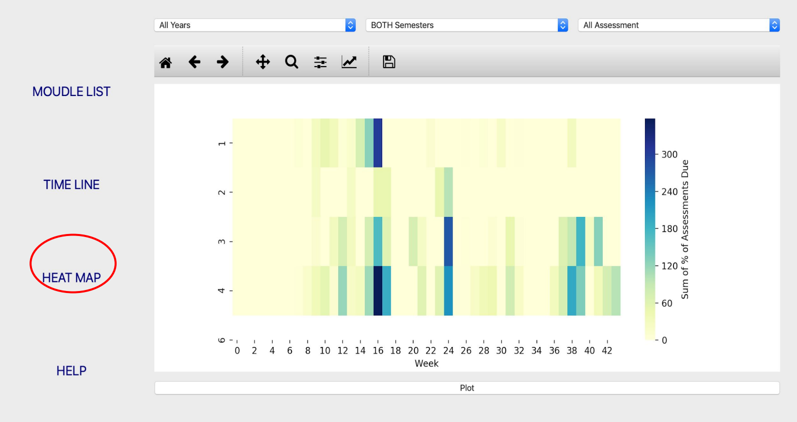
Academic Leader : name

Semester : Semester 1 (or 2) only

Assessment :

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

Heatmap sample



Early version of module booklet

Axure RP 8

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 ✔️

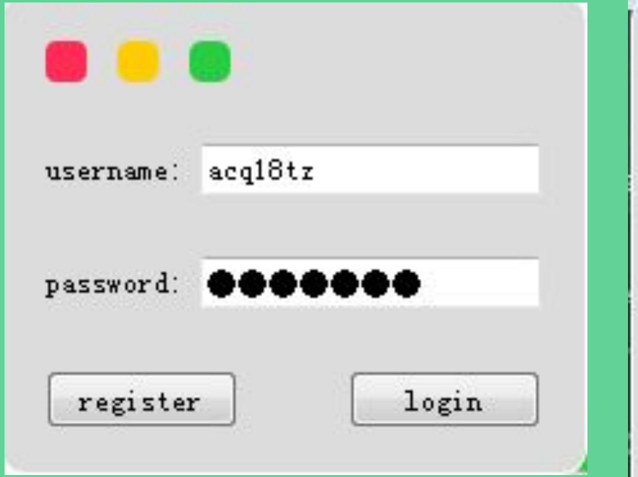
Sprint 3

Refining the interface

Sprint 3 User Stories

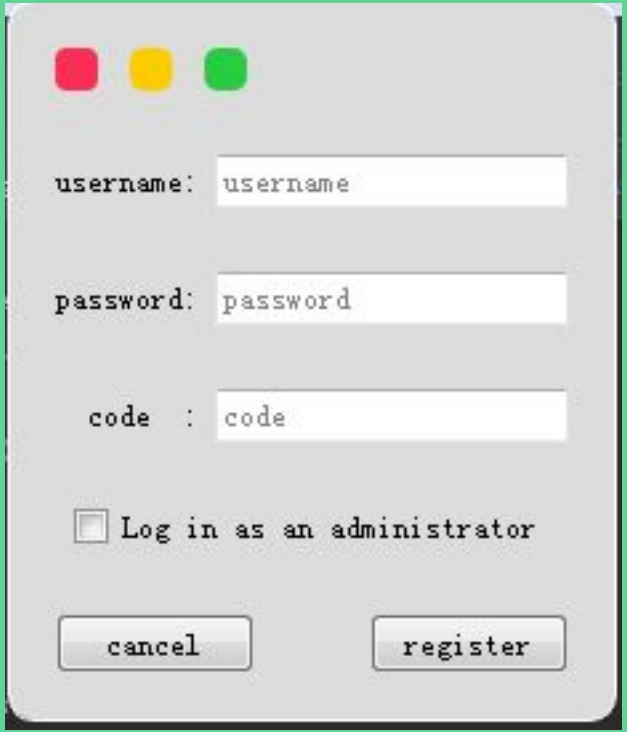
- As a user, I want a login and password protection, so that only verified colleagues can alter the data
- As a member of academic staff, I want to be able to enter information on my module's assessments, so that academic tutors will be able to account for my module
- As an LTM, I want to be able to 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



username: acq18tz

password: ●●●●●●●

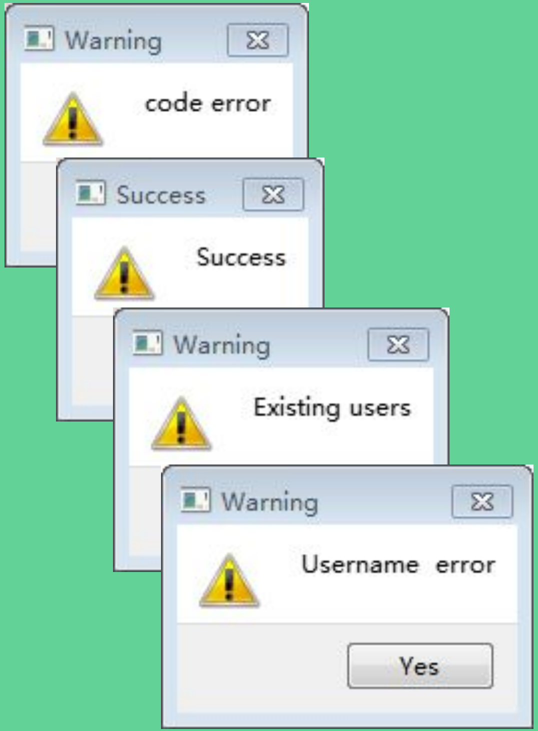


username: username

password: password

code : code

□ Log in as an administrator



Warning

code error

Success

Success

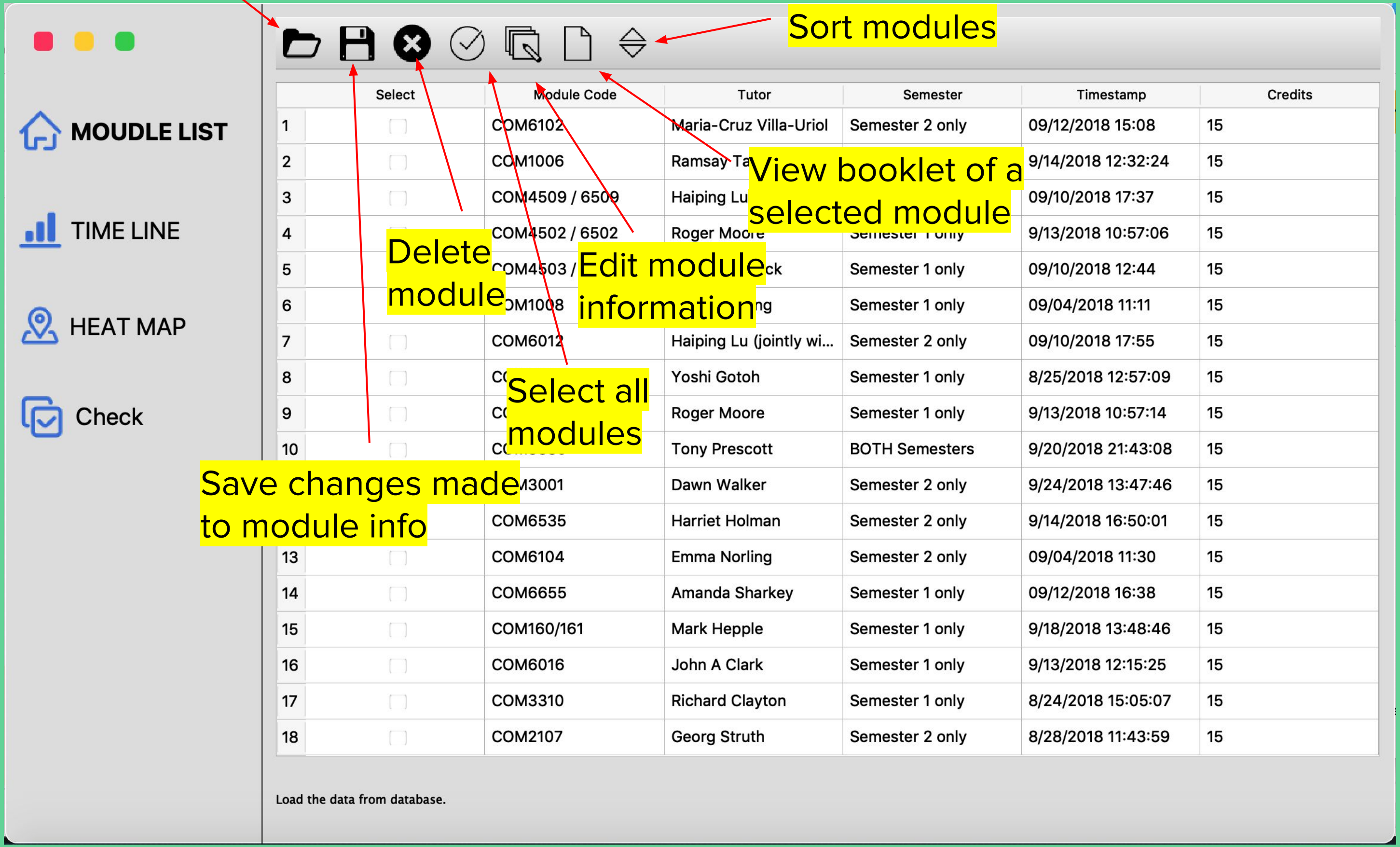
Warning

Existing users

Warning

Username error

Module list



Click to load data from CSV

Sort modules

View booklet of a selected module

Edit module information

Delete module

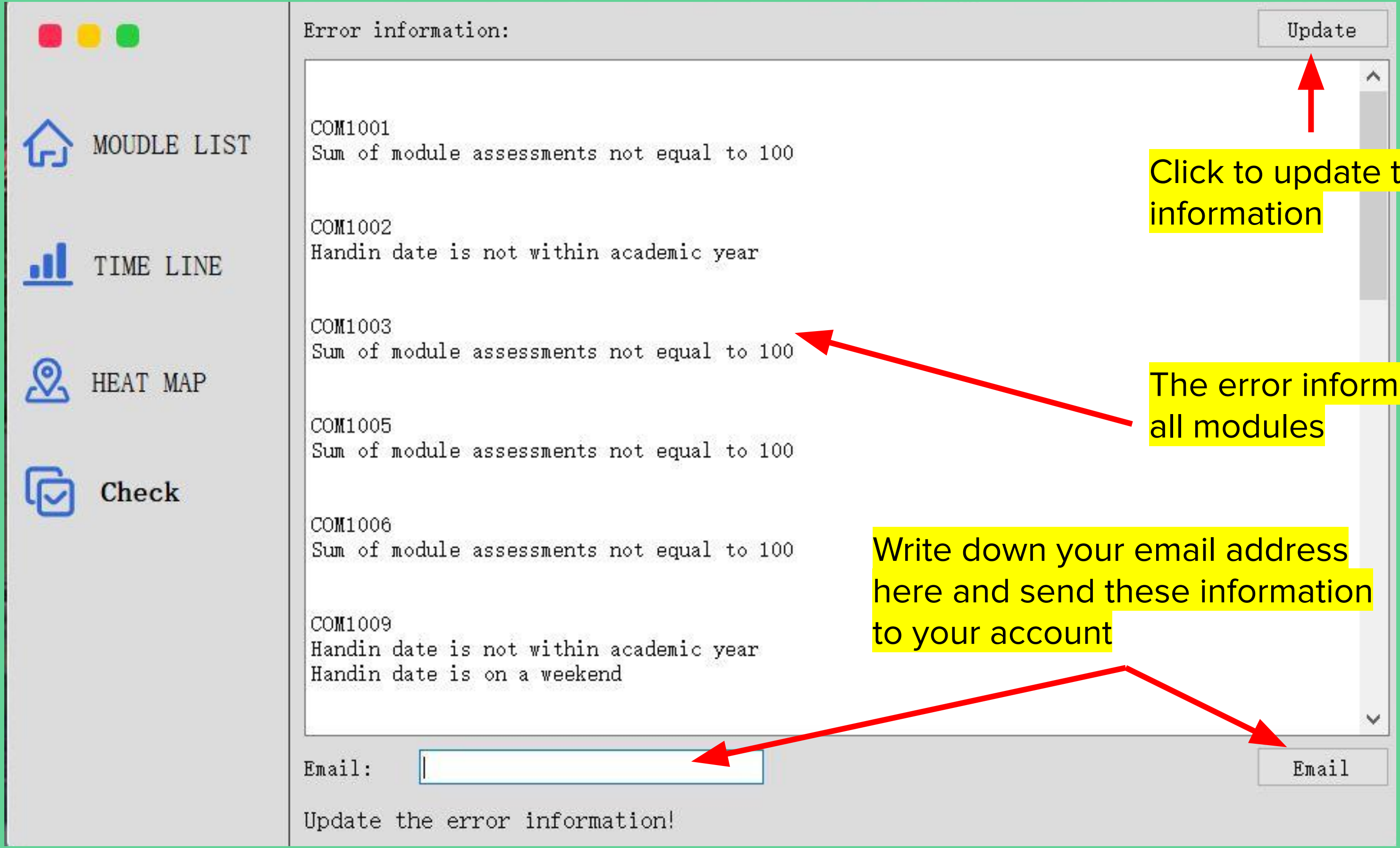
Select all modules

Save changes made to module info

	Select	Module Code	Tutor	Semester	Timestamp	Credits
1	□	COM6102	Maria-Cruz Villa-Uriol	Semester 2 only	09/12/2018 15:08	15
2	□	COM1006	Ramsay Ta		9/14/2018 12:32:24	15
3	□	COM4509 / 6509	Haiping Lu		09/10/2018 17:37	15
4		COM4502 / 6502	Roger Moore	Semester 1 only	9/13/2018 10:57:06	15
5		COM4503 /	ck	Semester 1 only	09/10/2018 12:44	15
6		M1008	ng	Semester 1 only	09/04/2018 11:11	15
7	□	COM6012	Haiping Lu (jointly wi...	Semester 2 only	09/10/2018 17:55	15
8	□	C	Yoshi Gotoh	Semester 1 only	8/25/2018 12:57:09	15
9	□	C	Roger Moore	Semester 1 only	9/13/2018 10:57:14	15
10	□	C	Tony Prescott	BOTH Semesters	9/20/2018 21:43:08	15
		M3001	Dawn Walker	Semester 2 only	9/24/2018 13:47:46	15
		COM6535	Harriet Holman	Semester 2 only	9/14/2018 16:50:01	15
13	□	COM6104	Emma Norling	Semester 2 only	09/04/2018 11:30	15
14	□	COM6655	Amanda Sharkey	Semester 1 only	09/12/2018 16:38	15
15	□	COM160/161	Mark Hepple	Semester 1 only	9/18/2018 13:48:46	15
16	□	COM6016	John A Clark	Semester 1 only	9/13/2018 12:15:25	15
17	□	COM3310	Richard Clayton	Semester 1 only	8/24/2018 15:05:07	15
18	□	COM2107	Georg Struth	Semester 2 only	8/28/2018 11:43:59	15

Load the data from database.

Check



Error information:

```
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:

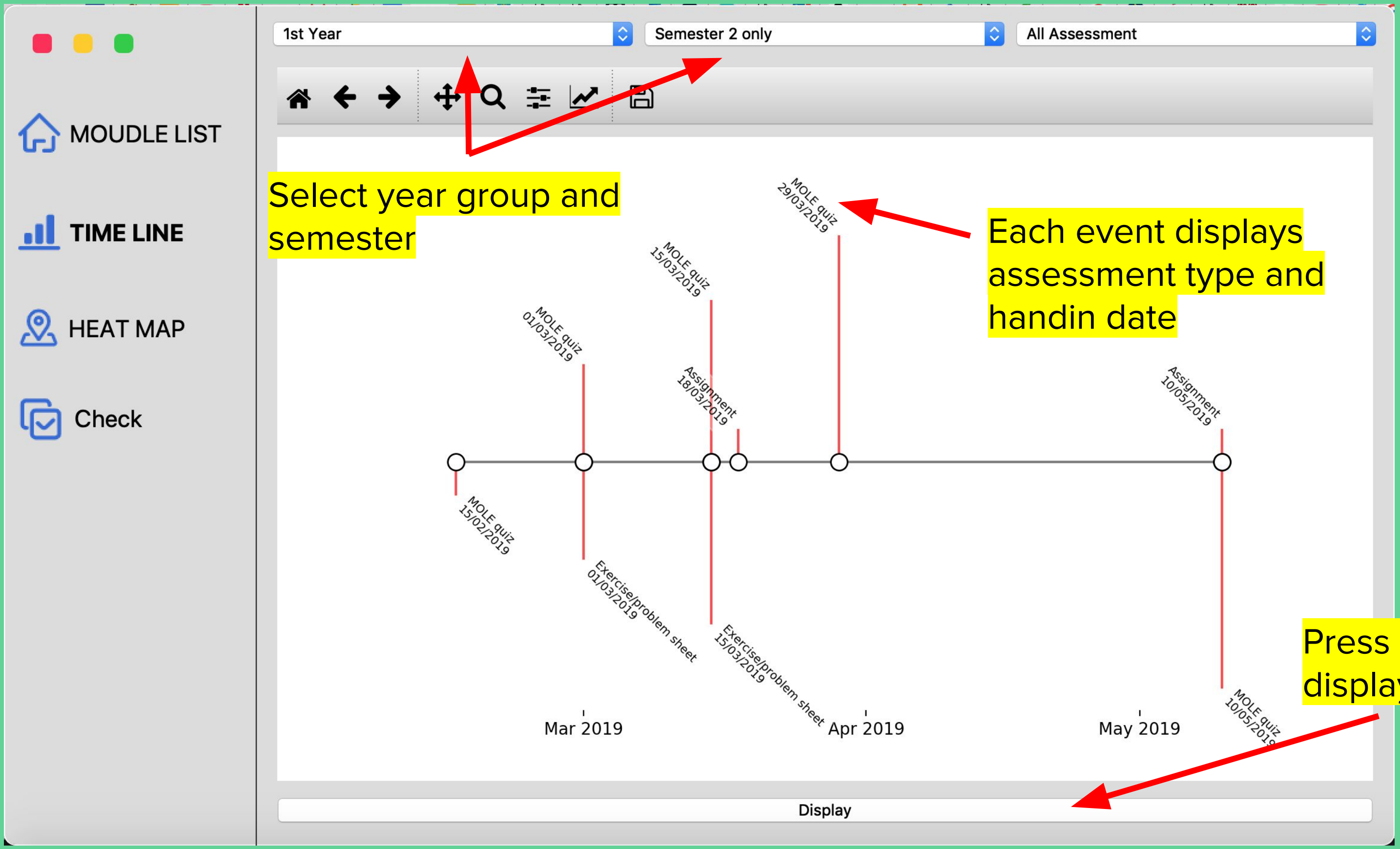
Update the error information!

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

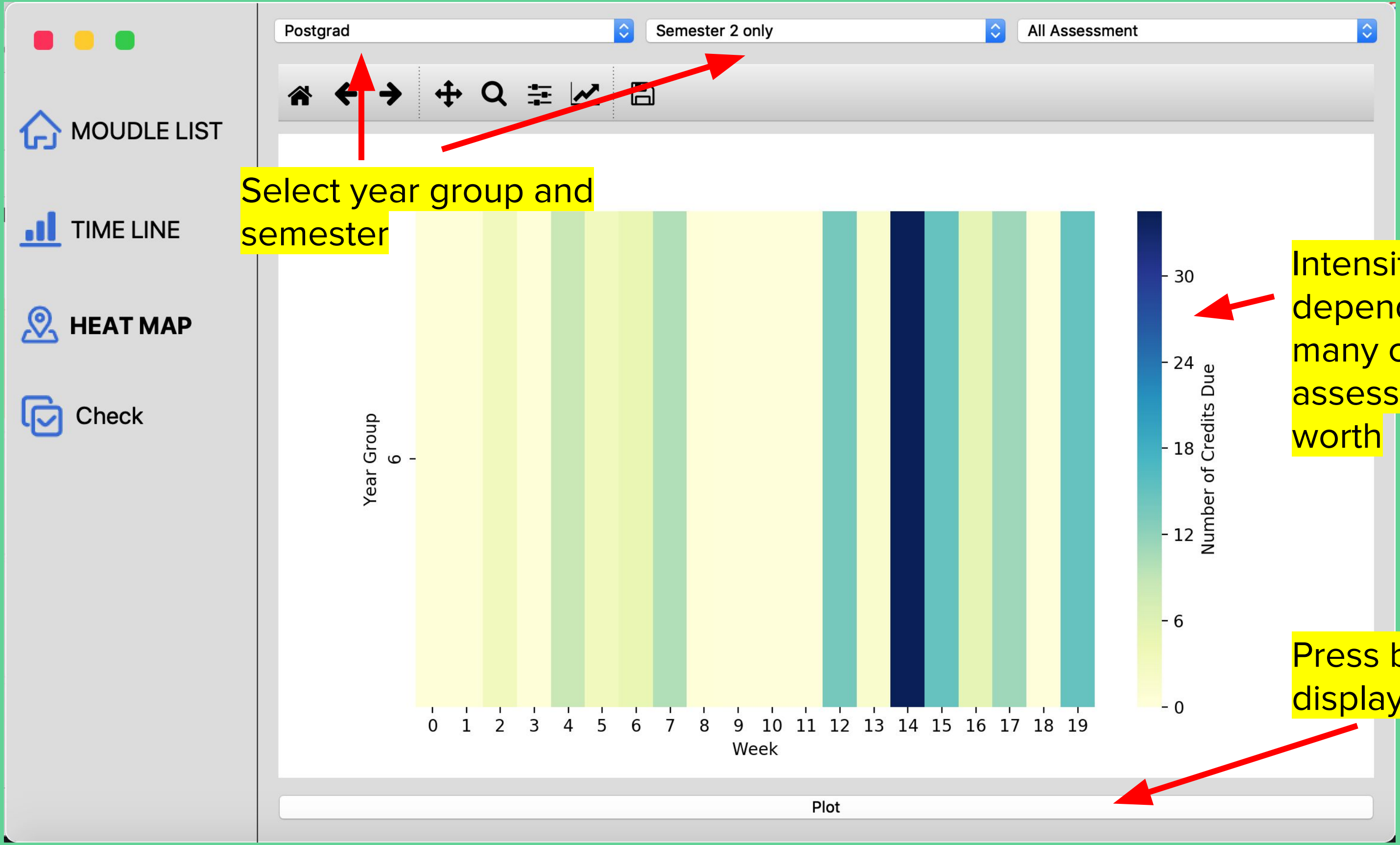


Select year group and semester

Each event displays assessment type and handin date

Press button to display timeline

Heatmap



Postgrad

Semester 2 only

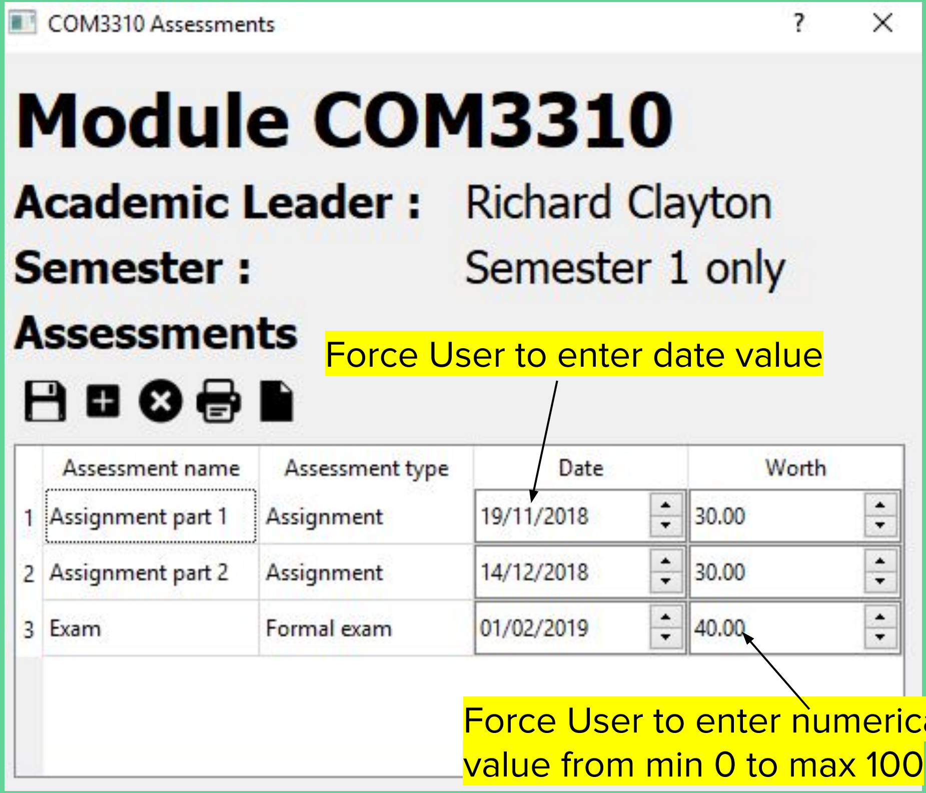
All Assessment

Select year group and semester

Intensity of color depends on how many credits each assessment is worth

Press button to display timeline

Final Booklet Format and Features



Module COM3310

Academic Leader : Richard Clayton

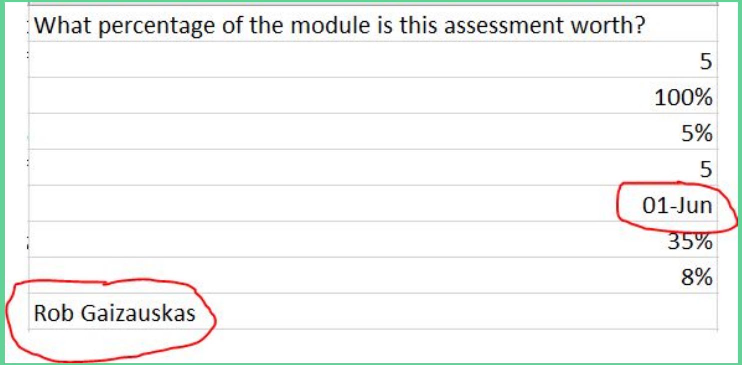
Semester : Semester 1 only

Assessments

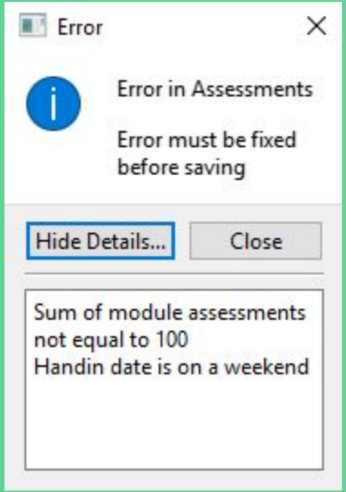
	Assessment name	Assessment type	Date	Worth
1	Assignment part 1	Assignment	19/11/2018	30.00
2	Assignment part 2	Assignment	14/12/2018	30.00
3	Exam	Formal exam	01/02/2019	40.00

Force User to enter date value

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 in Assessments

Error must be fixed before saving

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

Sprint 3 User Stories - completed?

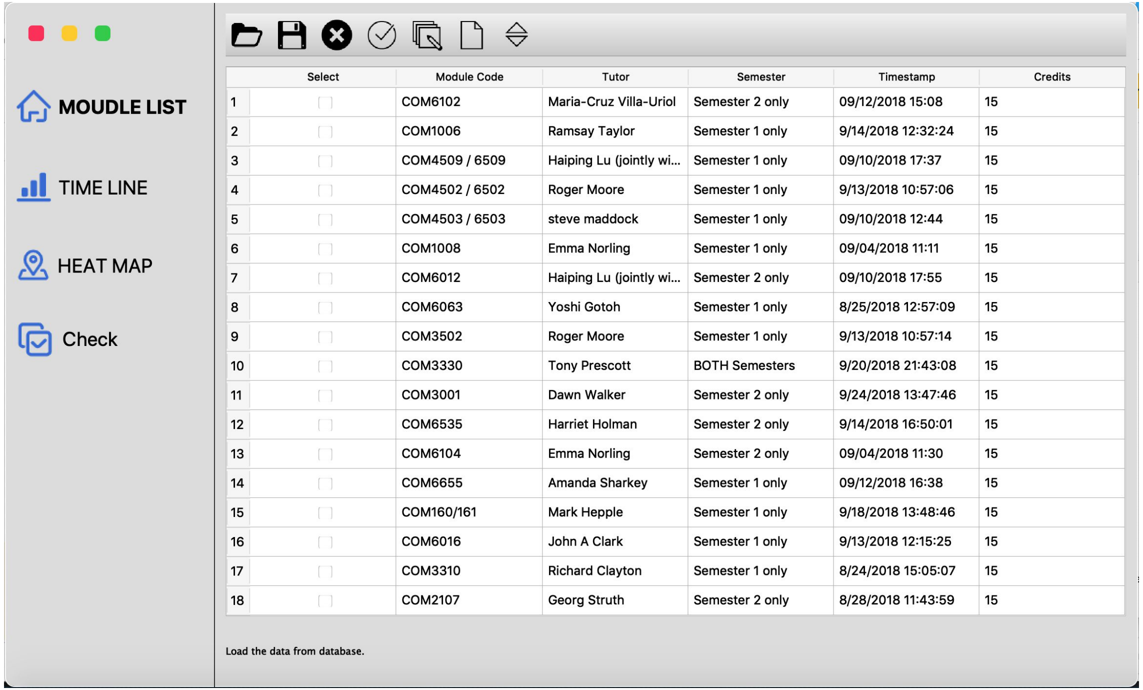
- As a user, I want a login and password protection, so that only verified colleagues can alter the data ✔️
- As a member of academic staff, I want to be able to enter information on my module's assessments, so that academic tutors will be able to account for my module ✔️
- As an LTM, I want to be able to add new login usernames and passwords, so that I can control who has access to the software ✔️
- As a 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 a administrative assistant, 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 ✔️

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!



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

Load the data from database.