

FRONT SHEET

BENS AND TOBY

COMP1531 ASSIGNMENT 2

Contributions		
Benjamin Pick	<z5162064>	33%
Benjamin Woodforde	<z5165086>	33%
Ziming(Toby) Zheng	<z5052592>	34%

User Stories

- Story Points are worth 2.5 hours
- Priority: 1-10, 10 for highest priority and 1 for lowest

	User Story Overview	
	User Story ID	Description
1	login	Students, staff, or guests can login and logout
2	gue_create	Guests are able to create an account
3	create_c	Staff can create a course
4	create_s	Staff can create a seminar
5	view_created	Staff can view events (courses and seminars) that they have created
6	edit_c	Staff can edit a course
7	edit_s	Staff can edit a seminar
8	view_open_c	Students, staff, or guests can view all 'open' courses
9	view_open_s	Students, staff, or guests can view all 'open' seminars
10	stu_reg_c	Students can register for a course
11	stu_reg_s	Students can register for a seminar
12	sta_reg_c	Staff can register for a course
13	sta_ref_s	Staff can register for a seminar
14	gue_reg_c	Guests can register for a course
15	gue_reg_s	Guests can register for a seminar
16	view_reg	Students, staff, or guests can view events (courses and seminars) they are registered in
17	deregister	Student, staff or guest can deregister from an event (courses and seminars)
18	cookies	Student, staff or guest can automatically be redirected to their dashboard if they logged in before within cookies existence time

ID: login
Name: Students, staff, or guests can login and logout
Description: As a student, staff or guest, I want to be able to login with unique authentication details, so that I can have my uniques information stored
Acceptance Criteria: <ul style="list-style-type: none"> • Student, staff or guest will have access to a page containing two text input boxes and a 'Login' button • Students and staff can login to the EMS using an zID and password • Guests can login to the EMS using an email and password • If the details are not correct or do not exist, the login page will remain. • After the details are authenticated, the page will redirect to 'open events' page.
Time Estimate (story points): 2
Priority: 9

ID: gue_create
Name: Guest users are able to create an account
Description: As a guest I want to be able to create unique authentication details, in the form of an email and password, so I can register to go to events (courses and seminars)
Acceptance Criteria: <ul style="list-style-type: none"> • A guest is able to visit an account creation page that has an input for an email and a password via a link provided on homepage. • A guest cannot submit a empty account form • A guest cannot input an invalid email format • A guest cannot have more than 40 characters of email • A guest cannot have more than 30 characters of password • A guest cannot used the email that has the same one as an already existing account
Time Estimate (story points): 2
Priority: 9

ID: create_c
Name: Staff can create a course
Description: As a staff member, I want to be able to create a course, so that the course exists on the system
Acceptance Criteria: <ul style="list-style-type: none"> • Staff members are able to access a form to enter information into text input boxes by pressing the 'Create Event' button and choosing 'course' on their 'Dashboard' • The information that will be required to be entered includes: <ul style="list-style-type: none"> ○ Course Name ○ Venue ○ Maximum Capacity

<ul style="list-style-type: none"> ○ Start and End Date ○ Start and End Time ○ End of Deregistration Period Date ○ Start and End of Early Bird Period ○ A description containing any extra plain text (pre reading, content, etc.) ● All input boxes will be required to have content ● Input boxes will have checks to ensure valid input ● Dates will have an authentication to ensure they are valid dates and times <ul style="list-style-type: none"> ○ Actual progression of event times ○ Formatting
Time Estimate (story points): 2.5
Priority: 8

ID: create_s
Name: Staff can create a seminar
Description: As a staff member, I want to be able to create a seminar, so that the seminar exists on the system
Acceptance Criteria: <ul style="list-style-type: none"> ● Staff members are able to access a form to enter information into text input boxes by pressing the 'Create Event' button and choosing 'Seminar' on their 'Dashboard' ● Staff have to add at least one session to the seminar by pressing an 'Add Session' button, which will add more input boxes to the page ● The information that will be required for the seminar includes: <ul style="list-style-type: none"> ○ Venue ○ Start and End Date ○ End of Deregistration Period Date ○ Start and End of Early Bird Period ○ Description containing any extra plain text information ● The information that will be required for each session includes: <ul style="list-style-type: none"> ○ Speaker ○ Venue ○ Maximum Capacity ○ Date ○ Start and End Time ○ Description containing any extra plain text information ● All input boxes will be required to contain information ● Input boxes will have checks to ensure valid input ● Dates will have an authentication to ensure they are valid dates and times <ul style="list-style-type: none"> ○ Actual progression of event times ○ Formatting ● Each session date will need to be within the seminar's start and end date
Time Estimate (story points): 2.5
Priority: 8

ID: view_created
Name: Staff can view events (courses and seminars) that they have created
Description:

As staff, I want to be able to view all 'open', 'closed' and 'cancelled' events (courses, seminars) that I have created on a 'Dashboard', so I can manage my created events.

Acceptance Criteria:

- Staff have a clickable 'Dashboard' navigation button
- Staff can view all 'open', 'closed' and 'cancelled' events that they have created (convened) on a 'Dashboard'
- If the staff has not created any events, display text reading something similar to: "There are no events that you have created."
- Events have a 'Toggle Details' button that will display an event's details.
 - Courses will display: course name, venue, convenor, capacity, start/end time, start/end date, deregistration period, early bird period, description, status, list of attendees
 - Seminars will display: seminar name, venue, convenor, start/end date, deregistration period, early bird period, description, status, list of sessions
 - Sessions will display: session name, date, start/end time, capacity, speaker, description, list of attendees

Time Estimate (story points): 2

Priority: 7

ID: edit_c

Name: Staff can edit a course

Description:

As a staff member, I want to be able to edit a course's information and change it's status, so that I can add new information to or cancel a a course

Acceptance Criteria:

- Staff members are able to press a course's 'Edit' button on the 'Dashboard' that will take them to a different form
- Staff members have access to a course's 'Cancel Event' button on their 'Dashboard' that will change the course's status to 'cancelled'
- The form will contain text input boxes for the following information:
 - Course Name
 - Venue
 - Maximum Capacity
 - Start and End Date
 - Start and End Time
 - End of Deregistration Period Date
 - Start and End of Early Bird Period
 - A description containing any extra plain text (pre reading, content, etc.)
- All input boxes will be required to contain information
- Input boxes will have checks to ensure valid input
- Dates will have an authentication to ensure they are valid dates and times
 - Actual progression of event times
 - Formatting

Time Estimate (story points): 2

Priority: 5

ID: edit_s

Name: Staff can edit a seminar

Description:

As a staff member, I want to be able to edit a seminar's information and change it's status, so that I can add new information to or cancel a seminar

Acceptance Criteria:

- Staff members are able to press a seminar's 'Edit' button on the 'Dashboard' that will take them to a different form
- Staff members have access to a seminar's 'Cancel Event' button on their 'Dashboard' that will change the course's status to 'cancelled'
- The form will contain text input boxes for the following information:
 - Course Name
 - Venue
 - Start and End Date
 - End of Deregistration Period Date
 - Start and End of Early Bird Period
 - A description containing any extra plain text (pre reading, content, etc.)
- The form will also contain a list of sessions that each have a 'Toggle Details' button that will show input text boxes for the respective session that includes:
 - Speaker
 - Venue
 - Maximum Capacity
 - Date
 - Start and End Time
 - Description containing any extra plain text information
- All input boxes will be required to contain information
- Input boxes will have checks to ensure valid input
- Dates will have an authentication to ensure they are valid dates and times
 - Actual progression of event times
 - Formatting

Time Estimate (story points): 2

Priority: 5

ID: view_open_c

Name: Students, staff, and guests can view all 'open' courses

Description:

As a student, staff or guest, I want to be able to view all 'open' courses, so I can choose courses to register in.

Acceptance Criteria:

- Users have a clickable 'Events' navigation button
- Users can view all 'open' courses on an 'Events' page
- Courses have a 'Register' or 'Deregister' button that will allow the user to register or deregister from an event
- Courses will have a 'Toggle Details' button that will display an events details
 - Name, venue, convenor, amount of positions left, start/end date, deregistration period, early bird period, description

Time Estimate (story points): 1.5

Priority: 8

ID: view_open_s

Name: Students, staff, and guests can view all 'open' seminars
Description: As a student, staff or guest, I want to be able to view all 'open' seminars, so I can choose events to register in.
Acceptance Criteria: <ul style="list-style-type: none"> • Users (students, staff, guests) have a clickable 'Events' navigation button • Users (students, staff, guests) can view all 'open' seminars on an 'Events' page • Seminars have a 'Register' or 'Deregister' button that will allow the user to view sessions that they can register or deregister from. • Seminars and their sessions will have a 'Toggle Details' button that will display the seminars details <ul style="list-style-type: none"> ◦ <u>Seminars</u> will display: name, venue, convenor, start/end date, deregistration period, early bird period, description, list of sessions ◦ <u>Sessions</u> will display: name, date, start/end time, amount of positions left, speaker, description
Time Estimate (story points): 1.5
Priority: 8

ID: stu_reg_c
Name: Students can register for open courses
Description: As a student, I want to be able to register for a open course, so that I can attend the course
Acceptance Criteria: <ul style="list-style-type: none"> • A student can click on a course's 'Register' button on the 'Events' page to attempt to register for a course • If the course is full, the student won't be able to register • If the registration is successful, a confirmation message "Register Success" is given • If the registration is unsuccessful, a declined message "Register Failed" is given • If the registration is successful, the 'Register' button will become a 'Deregister' button for the current student • If the registration is unsuccessful, the 'Register' button will remain the same. • If the registration is successful, the student will be added to the course's list of attendees
Time Estimate (story points): 3.5
Priority: 10

ID: stu_reg_s
Name: Students can register for open seminars
Description As a student, I want to be able to register for a open seminar , so that I can attend the seminar
Acceptance Criteria: <ul style="list-style-type: none"> • A student must choose at least one session from a list of sessions given for a seminar to register on the 'Events' page • If all the sessions are full, the student won't be able to register this seminar • If the registration of seminar is successful, a student has registered at least one session for this

seminar <ul style="list-style-type: none"> • If the registration of seminar is successful, a confirmation message "Register Success" is given • If the registration of seminar is unsuccessful, a declined message "Register Failed" is given • If the registration of seminar is successful, the 'Register' button will become a 'Deregister' button for the current student • If the registration of seminar is unsuccessful, the 'Register' button will remain the same. • If the registration of seminar is successful, the student will be added to the sessions' list of attendees respectively which he or she has successfully registered for.
Time Estimate (story points): 3.5
Priority: 10

ID: sta_reg_c
Name: Staff can register for open courses
Description: As a staff member, I want to be able to register for a open course, so that I can attend the course
Acceptance Criteria: <ul style="list-style-type: none"> • A staff member can click on a course's 'Register' button on the 'Events' page to attempt to register for a course • If the course is full, the staff member won't be able to register • If the staff member created the course(is convenor), they won't be able to register • If the registration is successful, a confirmation message "Register Success" is given • If the registration is unsuccessful, a declined message "Register Failed" is given • If the registration is successful, the 'Register' button will become a 'Deregister' button for the current staff member • If the registration is unsuccessful, the 'Register' button will remain the same. • If the registration is successful, the staff member will be added to the course's list of attendees
Time Estimate (story points): 3.5
Priority: 10

ID: sta_reg_s
Name: Staff can register for open seminars
Description: As a staff member, I want to be able to register for a open seminar, so that I can attend the seminar
Acceptance Criteria: <ul style="list-style-type: none"> • A staff member must choose at least one session from a list of sessions given for a seminar to register on the 'Events' page • If all the sessions are full, the staff member won't be able to register this seminar • If the staff member created the seminar(is convenor), they won't be able to register • If the registration of seminar is successful, the staff member has registered at least one session for this seminar • If the registration of seminar is successful, a confirmation message "Register Success" is given • If the registration of seminar is unsuccessful, a declined message "Register Failed" is given • If the registration of seminar is successful, the 'Register' button will become a 'Deregister' button for the current staff member • If the registration of seminar is unsuccessful, the 'Register' button will remain the same. • If the registration of seminar is successful, the staff member will be added to the sessions' list of

attendees respectively which he or she has successfully registered for.
Time Estimate (story points): 3.5
Priority: 10

ID: gue_reg_c
Name: Guests can register for open courses
Description: As a guest, I want to be able to register for a open course, so that I can attend the course
Acceptance Criteria: <ul style="list-style-type: none"> • A guest can click on a course's 'Register' button on the 'Events' page to attempt to register for a course • If the course is full, the guest won't be able to register • A guest will have to pay a registration fee \$10 to register for a course <ul style="list-style-type: none"> ◦ There will be an early bird period where the guest will have a 50% discount on registering for the course before a certain date • If the registration is successful, a confirmation message "Register Success" is given • If the registration is unsuccessful, a declined message "Register Failed" is given • If the registration is successful, the 'Register' button will become a 'Deregister' button for the current guest • If the registration is unsuccessful, the 'Register' button will remain the same. • If the registration is successful, the guest will be added to the course's list of attendees
Time Estimate (story points): 3.5
Priority: 10

ID: gue_reg_s
Name: Guests can register for open seminars
Description: As a guest, I want to be able to register for a open seminar, so that I can attend or speak at that seminar
Acceptance Criteria: <ul style="list-style-type: none"> • A guest must choose at least one session from a list of sessions given for a seminar to register on the 'Events' page • If all the sessions are full, the guest won't be able to register this seminar • If the guest is a speaker of a session in the seminar, he or she will be guaranteed to have a seat in that session • If the guest is not the speaker of a session in the seminar, he or she will pay a registration fee \$5 for this session • If the guest is the speaker of a session in the seminar, he or she won't pay any fee for the session • The amount of fee that the guest should pay for the seminar is the sum of fees of all sessions he or she successfully registered • If the registration of seminar is successful, the guest has registered at least one session for this seminar • If the registration of seminar is successful, a confirmation message "Register Success" is given • If the registration of seminar is unsuccessful, a declined message "Register Failed" is given • If the registration of seminar is successful, the 'Register' button will become a 'Deregister' button for the current guest • If the registration of seminar is unsuccessful, the 'Register' button will remain the same.

<ul style="list-style-type: none"> If the registration of seminar is successful, the guest will be added to the sessions' list of attendees respectively which he or she has successfully registered for.
Time Estimate (story points): 3.5
Priority: 10

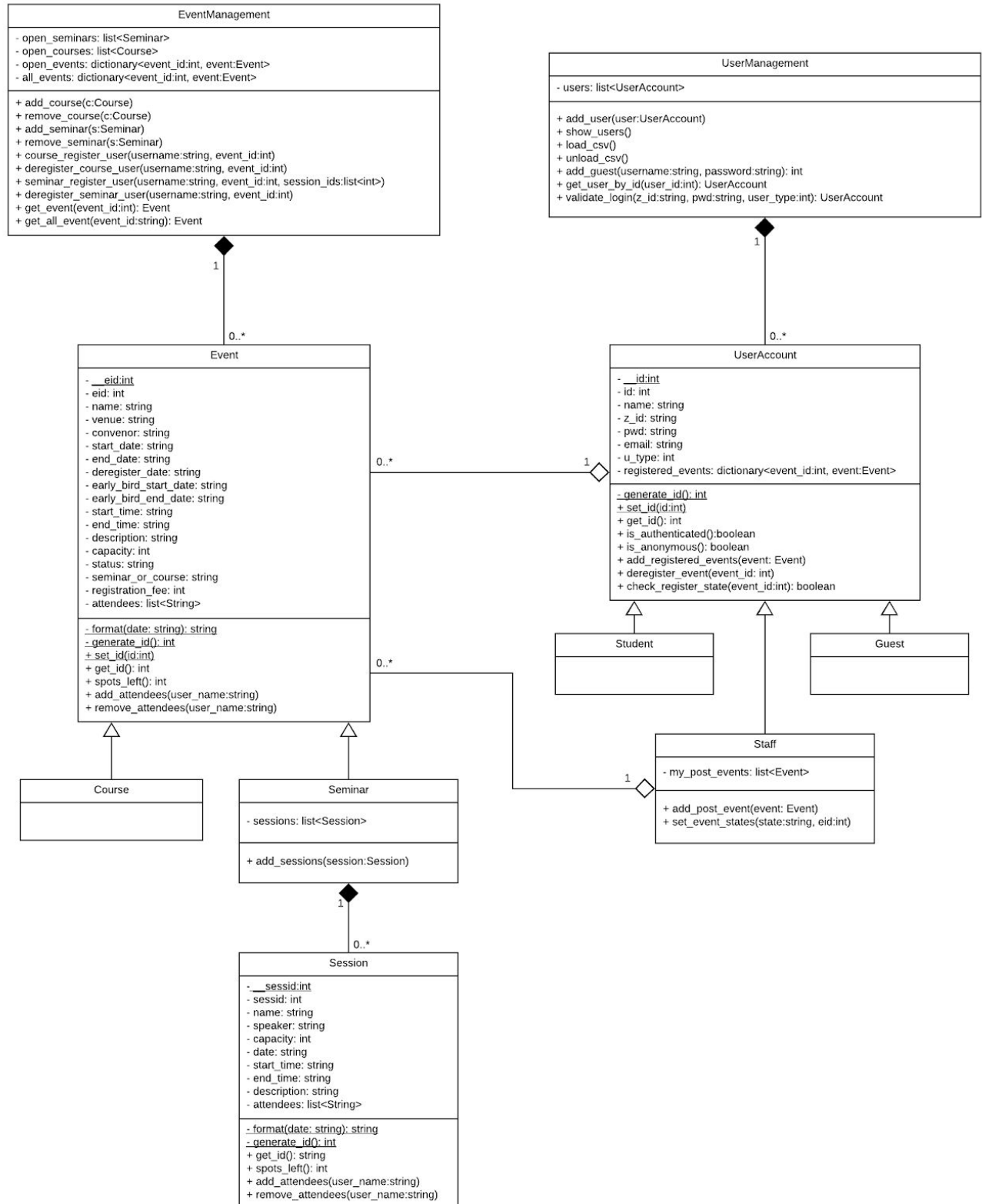
ID: view_reg
Name: Students, staff, or guests can view events (courses, seminars) they are registered in
Description: As a student, staff or guest, I want to be able to view all 'open', 'closed' and 'cancelled' events (courses, seminars) that I have registered in on a dashboard, so that I can manage my registration.
Acceptance Criteria: <ul style="list-style-type: none"> Students, staff, or guests have a clickable 'Dashboard' navigation button on 'Events' page Students, staff, or guests can view all 'open', 'closed' and 'cancelled' events (courses, seminars) that they have been registered to on a 'Dashboard' If an event is cancelled, there will be a red text message saying "This event is cancelled!" If the user has not registered in any events (courses, seminars), display text reading something similar to: "There are no events that you have registered in." Events (courses, seminars, sessions) have a 'Toggle Details' button that will display an event's details. <ul style="list-style-type: none"> <u>Courses</u> will display: course name, venue, convenor, capacity, start/end time, start/end date, deregistration period, early bird period, description, status <u>Seminars</u> will display: seminar name, venue, convenor, start/end date, deregistration period, early bird period, description, status, list of sessions <u>Sessions</u> will display: session name, date, start/end time, capacity, speaker, description
Time Estimate (story points): 2
Priority: 7

ID: deregister
Name: Students, staff, or guests can deregister from an event (courses and seminars)
Description: As a student, staff or guest, I want to be able to deregister from an event (courses and seminars), so that I can allow others to attend an event instead of me
Acceptance Criteria: <ul style="list-style-type: none"> A student, staff or guest can click on the 'Deregister' button on the 'Events' page to attempt to deregister from an event If the deregistration attempt is not in the deregistration period, the attempt will be unsuccessful If the deregistration is unsuccessful, a declined message "Deregister Failed" is given If the deregistration is successful, a confirmation message "Deregister Success" is given If the deregistration is successful, the 'Deregister' button will become a 'Register' button for the current student, staff or guest If the deregistration is unsuccessful, the "Deregister" button will remain the same If the deregistration of a course is successful, the student, staff or guest will be removed from the course's list of attendees If the deregistration of a seminar is successful, the student, staff or guest will be removed from all the registered sessions' list of attendees

Time Estimate (story points): 3
Priority: 9
ID: cookies
Name: Student, staff or guest can automatically be redirected to their dashboard if they logged in before within cookies existence time
Description: As a student, staff or guest, I want to go to my dashboard directly when I reopen the website after I closed it shortly so that I don't need to retype my username and password again in a short time
Acceptance Criteria: <ul style="list-style-type: none"> • If a student, staff or guest was at logged-in stage when they closed the website, the website will redirect to the 'Dashboard' page if they reopen it in a short time • The duration between closing and reopening is based on log-in cookies existence time in the browser
Time Estimate (story points): 1
Priority: 2

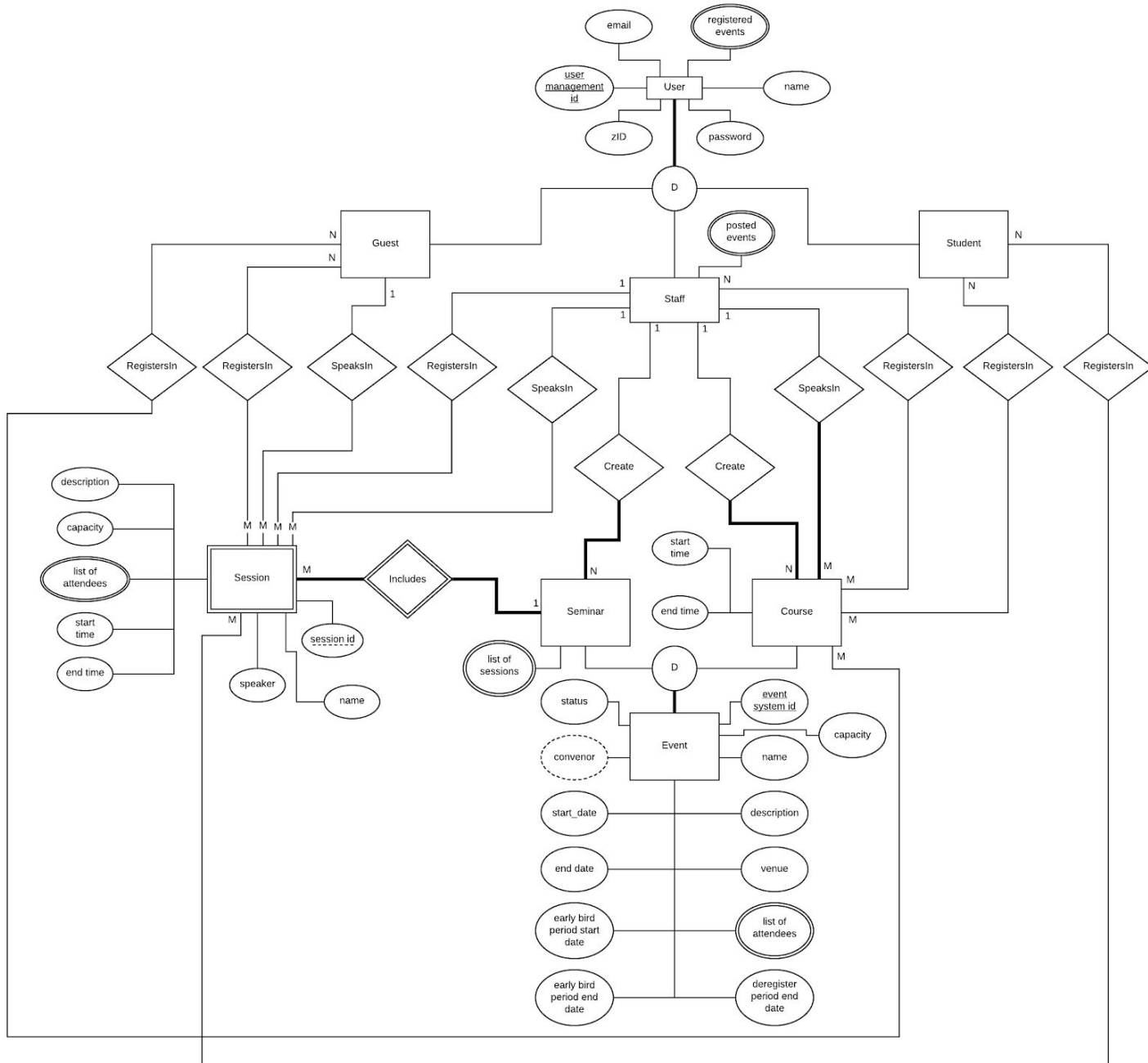
Class Diagram

(www.lucidchart.com/invitations/accept/60b8cf4c-2672-446b-94f8-b427fea522e1)



ER Diagram

(www.lucidchart.com/invitations/accept/f85284e3-8df7-4575-9f7e-f73abd5f2d87)



Log Book

Date of Meeting : 17/04/2018 Tuesday

- Summary of the meeting:
 - Week 07 Submission Schedule
 - Hopefully finish epic stories and user stories prior to Thursday's lab
 - Start doing UML class diagram during Thursday's lab
 - Review the work on Friday and submit it
 - Requirements Elicited:
 - In this meeting, we initially started thinking what the same behaviour of student and staff in general. As a conclusion, we think that
 - They are both register or deregister an event
 - They can login the system
 - They can view their dashboard in their account
 - They can view open events at the front page
 - We also consider staff have an exclusive behaviour. That is to create and edit events.
 - We have decided that the above behaviours consists of our epic story and each person has been assigned work to break down every epic story in details and create user story.
 - Key Design Ideas:
 - Identify different users (student and staff) of this system.
 - We can share the common functionalities in a broad view and can write in the epic story
 - our user story will be based on their different functionalities in different users
 - Identify the common functionalities of two types of user, and when we describe those functionalities, our epic story and user story will say "a user of the EMS"
 - We have identified some functionalities are in low priority. We may add these functionalities into user story on Thursday or Friday. They are listed as following:
 - Searching an event by name from a list of open events
 - Receiving a confirmation window
 - Viewing a list of attendees for an event
 - The rest are relatively in high priorities and we decided to think about them first between Tuesday and Thursday.
- Tasks to be accomplished next meeting:
 - To finish the epic story, and roughly finish user stories in high priorities
- Responsibilities for teammate
 - Ben P: Write this meeting's logbook and randomly pick one general behaviour to write user story

- Ben W: Rewrite our discussed epic stories in RGB format and pick one general behaviour to write user story
- Toby: Write user stories for the rest of general behaviours
- Some questions raised:
 - Can a user story have a word like “I shouldn’t be able to...”?
 - Updated on 18/04: No
 - Does attendee allow to book seminars or courses which have overlapping time?
 - Updated on 18/04: Since the spec doesn’t specify this, students or staff can have overlapping time

Date of Meeting: 19/04/2018 Thursday

- Summary of the meeting:
 - Reflection:
 - Ben P: I have almost finished writing user story but have questions to ask tutor. Hopefully, I can finish it in Friday afternoon
 - Ben W: I would like to have some feedbacks from tutors as well, otherwise I have done it
 - Toby: same as Ben P, I nearly finished the user story.
 - Plan for today and Friday before submission
 - Thursday during the lab:
 - Finish user story of high priority
 - Ask for some feedbacks for our temporary work
 - Review and modify our work based on the feedback
 - Drawing CRC cards and discuss what we need to put for each class
 - There will be a final meeting on Friday to review our work comprehensively and make sure everything is done correctly.
 - Requirements Elicited:
 - Epic story in low priority will be implemented after this meeting.
 - Only a staff can view a list of attendees
 - Both a student and a staff can receive confirmation window
 - We have decided that we would not write user story of searching an event by name because the spec doesn’t provide any information for that.
 - We have decided to replace “a user of the EMS” with “a student or staff”. This is because it is not a good style of writing, though the idea is the same.
 - Key Design Ideas: CRC cards for class diagram

EventManagerment	
Register a course/seminar	Student

Deregister a course/seminar Create courses/seminars Set state of an event Show current open courses Show current open seminars	Staff Course Seminar Registration
--	--

Course	
add attendees Remove attendees Show a list of attendees	EventManager

Seminar	
add attendees Remove attendees Show a list of attendees Add sessions	EventManager Session

Session	
Just store info of each session	Session

Registration	
Show confirmation	Event Manager Student Staff

Student	
authentication View dashboard	EventManager

Staff	
authentication View dashboard	EventManager

- Tasks to be accomplished next meeting:
 - Epic story and user story done

- UML class diagram roughly done
- Responsibilities for teammate
 - Ben P: Draw class diagram
 - Ben W: Write this meeting's logbook and finish up user story
 - Toby: Draw class diagram
- Some questions raised:
 - Is priority number unique for all the user stories?
 - Update on 19/04: No

Date of Meeting: 20/04/2018 Friday

- Summary of the meeting:
 - Reflection:
 - Ben P & Toby: working on class diagram
 - Ben W: review user story and epic story
 - Plan for Friday before submission
 - Clean up everything
 - Review everything
 - Submit the work
 - Week 07, Week 08 and Week 09 coding task schedule
 - Draw velocity chart
 - Key Design Ideas: Velocity Chart is used to monitor the progress
 - We have allocate three sprint reviews for the first iteration and another three sprint reviews for the second iteration
 - Tasks to be accomplished next meeting:
 - Finalise the submission
 - Responsibilities for teammate
 - Ben P: Draw class diagram
 - Ben W: Do the submission
 - Toby: Draw class diagram

Date of Meeting : 26/04/2018 Thursday

- Summary of the meeting:
 - The progress of everyones code was checked. This progress was then discussed and new task assignment was completed.
 - A lot of the meeting time was spent on working on the current weeks lab, so code checking was brief and the meeting acted as a reminder to complete any unfinished code.
 -
- Tasks to be accomplished next meeting:
 - To further complete any assigned code work
 - Continue developing classes as well as start to implement in html format
- Responsibilities for teammate
 - Ben P: Continue with his html implementation

- Ben W: Complete the logbook for this meeting, as well as implement the event management class in an html format
- Toby: Complete his originally assigned parts as well as compile other team members code into his desired format
- Some questions raised:
 - Are we allowed to use JavaScript?
 - Yes, but it has to work no matter what device is used

Date of Meeting : 03/05/2018 Thursday

- Summary of the meeting:
 - The progress of everyones code was checked. This progress was then discussed and new task assignment was completed.
 - Issues between communication in terms of coding styles and the overall project layout have been discovered and are being ironed out.
- Tasks to be accomplished next meeting:
 - To further complete assigned code work
 - Continue implementing the classes in html format
 - Complete some refactoring in order to maintain a constant coding style
- Responsibilities for teammate
 - Ben P: Continue with his html implementation
 - Ben W: Complete the logbook for this meeting, as well as implement the event management class in an html format
 - Toby: Complete his originally assigned parts as well as compile other team members code into his desired format
- Some questions raised:
 - How do you store information?
 - Toby is looking into it and implementing pickle into our project

Date of Meeting : 10/05/2018 Thursday iteration 1 submitted

- Summary of the meeting(Sprint Review 3):
 - Sprint Review 1 & 2 were not very successful because of group conflicts
 - Discussed what was not going well in terms of teamwork and workload splitting
 - Created ideas and started to implement these ideas of how to understand better
 - Discussed where Toby's work is up to
 - Discussed issues Toby got
 - Noted down things that could be changed to make them better as well as things that were not planned to be added but would be a valuable contribution
 - Below is the velocity chart for the iteration 1:
 - Grey is predicted workload and Blue one is actual completed workload



- Tasks to be accomplished next meeting:
 - Continue to work on the code for the project as well as prepare progress reports to be able to show an overseer
- Responsibilities for teammate
 - Ben P: Brainstorm and write down or implement any team benefitting ideas to discuss in next meeting
 - Ben W: Brainstorm and write down or implement any team benefitting ideas to discuss in next meeting
 - Toby: Brainstorm and write down or implement any team benefitting ideas to discuss in next meeting

Date of Meeting : 11/05/2018 Friday

Summary of events:

- Discussed progress with issues raised on github
- Discussed user stories completion - any bugs or non started
- Times
 - How to establish deregister date - Have convenor set date and time
 - Need time for sessions
- Screen Layout
 - Original screen: Title + login button
 - Dashboard hyperlinks into a toolbar-ish form
- Finding number of attendees by getting the length of the list
- Raised registering when a course is full - can't be allowed
- Want to finish fixing all raised issues by Wednesday - 16th May

Work Done:

- BP

- All Date Input Validation
- Field requirements have validation
- Convenor information is automatically passed when creating events
- Single selection radio buttons validated
- Checks if capacity is a reasonable number
- TZ
 - Fixed attendees bug
 - Check date format on CSE
 - Check cancelling an event
- BW
 - Fixing user stories, class diagram and logbook

Work Allocation

- BP
 - Still fixing other date issues
 - Adding times to courses and sessions (seminar is based off sessions)
 -
- TZ
 - Implementing pickle inside the models so data is saved properly
 - Loading data through pickle
- BW
 - We should have more categories of past events for a staff. For example, a staff can see current open course and he's created in the first two categories, course and seminars he made cancelled in the third and fourth categories and closed courses and seminars in the fifth and sixth categories.
 - We should have more categories of registered events for a staff/student. For example, a person can see my current registered course in the first category, my current registered seminar in the second category, my cancelled registered course in the third, my cancelled registered seminar in the fourth, my closed registered course in the fifth and my closed registered seminar in the sixth.
 - Creating HTML/CSS
 - When registering for an event, clicking the register button refreshes the page so what was previously expanded gets unexpanded

Date of Meeting: 17/5/2018

- Summary of the meeting:
 - Matters Discussed:
 - New Iteration Discussion
 - User Stories
 - Work distribution
 - Add a return to home page button in the login page
- Improvement of iteration 1:
 - Moving the deregister button to the dashboard

- The register button should just become unavailable
 - Need to add some CSS effects, and bootstrap is a good option
 - Need to add more CSS effects on the webpage
- Requirements Elicited for iteration 2:
 - Add a new user account - guest
 - Guest need to create an account first
 - Guest has registration fee if he or she is not the speaker of the session
 - We would like to predefined the fee for classes and sessions
 - Change the way of registering a seminar, must support registering a session

Work:

- Auto scroll down on click - javascript
- Cancel when entering info on creating an event

Ben P:

- For guest dashboard - modify event page to just have their info up the otp

Ben W:

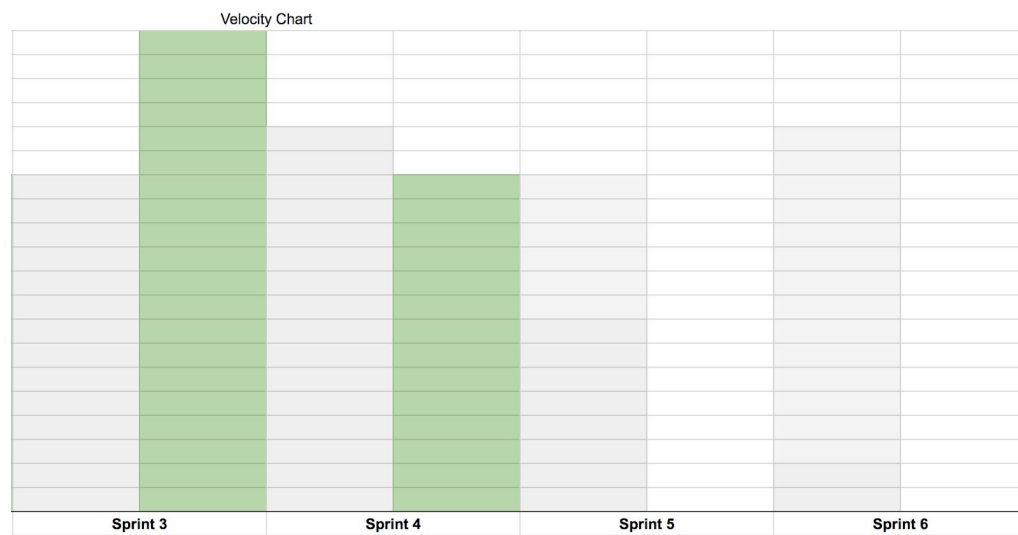
- ER Diagram
- Class Diagram
- User Stories
- Footnote/Footer
- Editing an event
- For dashboard pages - events that have been registered in and then been closed

Toby:

- Changes

Date of Meeting: 20/5/2018

- Sprint Review 4: velocity chart so far
 - Grey is predicted and blue is actually completed



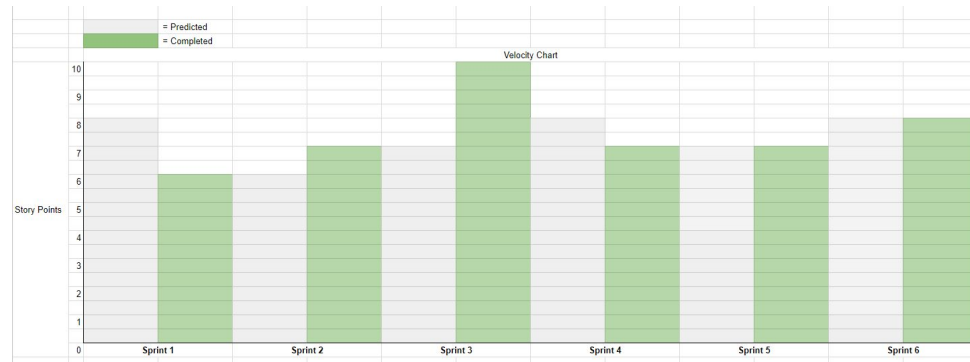
- Summary of the meeting:
 - Reflection of progress for iteration 2

- Date of Meeting:23/5/2018

-
- | Sprint | Number of Tasks Completed |
|----------|---------------------------|
| Sprint 4 | 7 |
| Sprint 5 | 6 |
| Sprint 6 | 8 |

- Date of Meeting: 26/5/2018

- Sprint Review 6:
 - Final Velocity Chart



- Remaining things:
 - Testing
 - CSS effects

Reflection

(Individual - so 3 reflections maybe 100-200 words long)

Ben Pick -

I believe that our work ethics and work motives changed a lot throughout the course of our project. Before iteration 1, we exhibited bad communication and vague aims, where a less finished version of our system was released and exhibited in the iteration 1. However, after discussion, our team set aside their differences and worked well together with clear goals, where every member was set aside concrete work, adapted to each person's skills. This made our working environment much easier, for example, I ended creating a lot of the javascript and some html and flask, where no one else in my group had studied much javascript. If we had to re-implement the system, I would have tested the system on many different machines, to remove the date-time inconsistencies between machines, resulting in some errors during our presentation. The addition of a list of speakers to choose from when creating an event could also be included as well as persistent storage of events, rather than just stored in cache, allowing for easier input of events. Using more flask, instead of javascript for error checking, could have also resulted in an easier testing on the python side, where try-except statements and better pytest could have been implemented.

Ben Woodforde

In relation to our teamwork and collaborative environment, it was far from perfect. We started off with little to no communication and it led to a lack of work being completed as well as a badly split workload. After the first iteration, we realised we had to pull our heads in and our communication became better but how we split work and ensured everyone did their work was still a little bit shaky, with the only point of reference being memories and an online chat room. How we could have improved our teamwork is by establishing spreadsheet or document where we assigned user stories or the amount of story points, and then recording the date when an objective is complete and the time it took to complete it.

In terms of the actual coding project, I personally struggled with the python and flask implementation at first, not really understanding how everything worked together, learning through the exercises in the labs. It would've been nice to have finished all of the labs before needing to start the coding. Our website ended up working quite well due to Pick's javascript ability and Toby's use of pickle. We had issues in our presentation with different browsers and how that affected CSS inputs and javascript implementation. If I were to repeat the project again with a different team knowing what I know now, I would've stored event information in a csv file as well as use more html templates instead of having everything on one dashboard page.

Toby Zheng

In terms of modelling classes, I should have considered performance (time complexity and space complexity) and discussed with teammates. For example, it's better to use all dictionaries to store both event data and user data. It's not only easy to deal with in Python but also avoid some redundant loops. Especially, in our current model, we have many mixed data structures like

half list and half dictionary. If I could re-implemented, I would discuss it beforehand and use one type of data structure consistently.

Also dealing with “date” format was a little messy in routes.py and sometimes my solution was ad-hoc. This was largely because I couldn’t have enough time to overhaul my code and coordinate with my teammates. If I could re-implemented, I would consider the flow thoroughly and implement in a systematic way.

Some extensions I can add on this project would be using pickle to store users’ and events’ data. About teamwork, I should have communicated with them more to avoid group conflicts.