

FINAL REPORT

Team name: Hac-kers

Team Members

Henry Pan (z5014389)

Chen Zhou (z5122526)

Austin Lam (z5116890)

Contribution (out of 100%)

Henry Pan: 40%

Chen Zhou: 30%

Austin Lam: 30%

Table of Contents

<i>Main Contribution for each.....</i>	<i>2</i>
<i>List of User Stories with Acceptance Criteria.....</i>	<i>3</i>
<i>Class Diagram.....</i>	<i>14</i>
<i>Entity Relationship Design.....</i>	<i>15</i>
<i>LOGBOOK</i>	<i>16</i>
<i>Authentication Details.....</i>	<i>23</i>
<i>Reflection</i>	<i>23</i>

Main Contribution for each

Henry:

1. Project setup
2. UML design and implementations
3. database ER design and implementation
4. index page
5. reusable component 'list event' to be used in multiple pages
6. user management(user login/logout, guest signin)
7. profile page
8. exception handling and security improvement
9. majority of front-end contribution

Austin:

1. Dashboard
2. Closing, cancelling and reopening events
3. Forgot my password page and reset password functionality
4. Unit tests
5. Separating events into attending, passed, cancelled, hosting and closed
6. Epic stories, user stories and acceptance criteria

Chen:

1. Initial repo setup
2. Initial uml class diagram
3. Event addition front/backend
4. Photo uploads
5. Basic Validation
6. user API endpoint
7. UX Tweaks

List of User Stories with Acceptance Criteria

1 story point = 2hrs

priority = 10 is highest priority, 1 is lowest priority

Epic stories are underlined

ES1 - As a UNSW student or staff, I should be able to view events on the website so that I can easily manage them.

US1 - As a UNSW student/staff, I should be able to log in with my login details so that I can view my student/staff account.

Acceptance Criteria:

- The website should present 2 textbox for input, one for zID and one for password so that the user can enter in their login details.
- The website should have a “forgot my password” link if the customer has forgotten their password and provide instructions to reset their password.
- The website should display “incorrect zID or password” if the user inputs an incorrect zID or password in the zID or password fields.
- The website must retrieve the logged in user’s account with their relevant account information when the user has successfully logged in with their zID and password.
- When the user types in their password incorrectly 5 times, the account of that zID will be locked and a message will display stating “account of zXXXXXXX has been locked. Please press “forgot my password” to reset your password.”

Estimate: 3 Story points

Priority: 9

US2 - As a UNSW student/staff, I should be able to log out of my student account so that no one can accidentally log back in.

Acceptance Criteria:

- The website must have a link for the user to logout on all pages of the website.
- Once the user has clicked the logout link, the website should display “logged out successfully” and redirect to the login page.

Estimate: 2 Story points

Priority: 9

ES2 - As a UNSW student or staff, I should be able to see all open seminar events and all open course events so I can easily choose from them.

US3 - As a UNSW student/staff, I should be able to see a list of all open seminar events on the homepage so that I will be able to find a seminar that most interests me.

Acceptance Criteria:

- When the user is viewing the homepage of their account, the user must be able to see a list of all open seminar events under a heading titled “Open Seminars”.

- Each seminar element within the list must be of the same size and must show the title of the event, a picture relating to the event, snippet of the description, the names of the speakers, number of spots available, place and the dates to be held.
- The website must show all open seminars within the list and must be scrollable if the list is long.

Estimate: 3 Story points

Priority: 6

US4 - As a UNSW student/staff, I should be able to see a list of all open course events on the homepage so that I will be able to find a course that most interests me.

Acceptance Criteria:

- When the user is viewing the homepage of their account, the user must be able to see a list of all open course events under a heading titled “Open Courses”.
- Each course element within the list must be of the same size and must show the title of the event, a picture relating to the event, snippet of the description, the names of presenter, number of spots available, place and the date to be held.
- The website must show all open courses within the list and must be scrollable if the list is long.

Estimate: 5 Story points

Priority: 7

ES3 - As a UNSW student or staff, I should be able to view more details of a particular event and register for it so that I can attend

US5 - As a UNSW student/staff, I should be able to click on a particular event so that I can read about what the event is about.

Acceptance Criteria:

- When the user clicks on an event, the user must be directed to the event page which must show the event’s information. I.e. title of event, picture, description, name of speakers, number of positions available, place and dates.
- The page must be updated so that it should show if the user is attending or can register for the event.
- If the user is attending and is still within the de-registration window, the de-register button should be displayed.
- The page should have links that can direct back to the homepage or to the user’s dashboard and also a link to log out.
- The system should identify the user type and staff/student should not pay for attending.

Estimate: 3 Story points

Priority: 6

US6 - As a UNSW student/staff, I should be able to register for the event if I am interested in it so that I can secure my reservation.

Acceptance Criteria:

- The page of the event must show the register button at the bottom of the page if there is enough spots available and if the user is not the convenor.
- Once the user clicks register, a confirmation message must be displayed stating “Your reservation is completed, a confirmation will be sent to your email address”.
- The number of available spots must update after user clicks register.
- The de-register button must show up for the user after user has registered and must show how long the de-register window will last for.
- The register button must disappear after the user has registered or the event has no more spots available.
- The page should have links that can direct back to the homepage or to the user’s dashboard and also a link to log out.

Estimate: 4 Story points

Priority: 5

US7 - As a UNSW student/staff, I should receive a confirmation after registering for an event so that I know I have completed a reservation.

Acceptance Criteria:

- The Website must retrieve the user’s email address and send a confirmation of attendance to the specified event with its details after the user has clicked register.

Estimate: 1 Story points

Priority: 5

US8 - As a UNSW student/staff, I should be to cancel my reservation within a specified de-registration window if I am not interested anymore

Acceptance Criteria:

- A message must be displayed after the user presses deregister stating “You have cancelled your attendance for this event.” and an email must be sent to the user to notify their cancellation of the event.
- The number of spots available for the event must update after the user has de-registered and the register button must appear again if there are still spots remaining.

Estimate: 1 Story points

Priority: 7

ES4 - As a UNSW student, I should be able to view all current and past events that I have registered for so that I can manage what events I will attend or have attended.

US9- As a UNSW student, I should be able to navigate to my dashboard to view my current events so that I know what I have registered for.

Acceptance Criteria:

- The page of the dashboard must show a section dedicated to all the current events that the user has registered for.
- Each event must be of the same size, must indicate whether it is a course or a seminar, must show the title of the event, a picture relating to the event, snippet of the description, the names of presenters, place and the dates to be held.
- Each event must have clickable title that will direct the user to the event page.
- The dashboard must have links on the top of the page to direct the user to log out or back to the homepage.
- The webpage must be scrollable if there are a lot of events for the page to display at once.
- Each event should display the number of seats available in the dashboard.

Estimate: 3 Story points

Priority: 7

US10 - As a UNSW student, I should be able to navigate to my dashboard to view my past events so I can see which events I have attended.

Acceptance Criteria:

- The page of the dashboard must show a section dedicated to all the past events that the user has registered for.
- Each past event must be of the same size, must indicate whether it was a course or a seminar, must show the title of the event, a picture relating to the event, snippet of the description, the names of presenters, place and the dates that it was held.
- Each event must not have clickable title to direct a user to a past event page.
- The dashboard must have links on the top of the page to direct the user to log out or back to the homepage.
- The webpage must be scrollable if there are a lot of events for the page to display at once.

Estimate: 2 Story points

Priority: 5

ES5 - As a UNSW staff, I should be able to view all current, past and cancelled events that I have registered or posted so that I can manage what events I will attend, have attended and those I have cancelled.

US11 - As a UNSW staff, I should be able to navigate to my dashboard to view my current events so that I know what I have registered for.

Acceptance Criteria:

- The page of the dashboard must show a section dedicated to all the current events that the user has registered for.

- Each event must be of the same size, must indicate whether it is a course or a seminar, must show the title of the event, a picture relating to the event, snippet of the description, the names of presenters, place and the dates to be held.
- Each event must be have clickable title that will direct the user to the event page.
- The dashboard must have links on the top of the page to direct the user to log out or back to the homepage.
- The webpage must be scrollable if there are alot of events for the page to display at once.
- Each event should display the number of seats available in the dashboard.

Estimate: 3 Story points

Priority: 7

US12 - As a UNSW staff, I should be able to navigate to my dashboard to view my past events so I can see which events I have attended.

Acceptance Criteria:

- The page of the dashboard must show a section dedicated to all the past events that the user has registered for.
- Each past event must be of the same size, must indicate whether it was a course or a seminar, must show the title of the event, a picture relating to the event, snippet of the description, the names of presenters, place and the dates that it was held.
- Each event must not have clickable title to direct a user to a past event page.
- The dashboard must have links on the top of the page to direct the user to log out or back to the homepage.
- The webpage must be scrollable if there are alot of events for the page to display at once.

Estimate: 2 Story points

Priority: 5

US13 - As a UNSW staff, I should be able to navigate to my dashboard to view my cancelled events so I can see which events I have cancelled.

Acceptance Criteria:

- The page of the dashboard must show a section dedicated to all the cancelled events that the user had registered for.
- Each cancelled event must be of the same size, must indicate whether it was a course or a seminar, must show the title of the event, place and the dates that it was to be held.
- The dashboard must have links on the top of the page to direct the user to log out or back to the homepage.
- The webpage must be scrollable if there are alot of events for the page to display at once.

Estimate: 1 Story points

Priority: 4

ES6 - As a UNSW staff, I should be able to be the convenor for my event and post either a seminar or course event online so I can inform people of my event.

US14 - As a UNSW staff, I should be able to create a seminar event online so that I can attract more attendees.

Acceptance Criteria:

- From the homepage of the staff user, there must be a link to create an event which will direct the user to an event creation page.
- The event creation page must have different fields that must be filled in by the user.
- When the user clicks on “Create Event”, there must be a field stating “Type of event” with a drop-down list that must list the 2 options (Course or Seminar) where the user can select the type of event they want.
- There must be a field stating “Title for event” with a text box for input so that the user can type in the title for their event.
- There must be a field stating “address” with a long textbox so that the user can enter in the address of the event.
- There must be a field stating “number of presenters/guest speakers” with a drop-down list of numbers of presenters. The number chosen will determine the number of text boxes that will be generated for the user to enter in the names of the presenters/guest speakers.
- There must be a field stating “de-register window time frame” with 3 drop-down lists ranging from “days”, “hours” and “minutes” so that the user can select how long the time frame can be.
- There must be a field stating “Early Bird” with a text box so that convenor can enter a number for the number of early bird days.
- When the user has filled in all fields, there must be a button at the bottom displayed as “Create Event” so that the user can click and post the event to the website.
- If there are fields missing when the user clicks “create event”, there must be an error message stating all the fields that have been missed at the top of the page. E.g. “Please fill in the title.”
- When the user clicks on “create event” with all fields correctly filled in, there must be a message stating “You have successfully created your event” and the user will be redirected to their homepage.
- A confirmation email message must be sent out to the user after creating their event with the relevant information for their event.

Estimate: 5 Story points

Priority: 9

US15 - As a UNSW staff, I should be able to create a course event online so that I can attract more attendees.

Acceptance Criteria:

- From the homepage of the staff user, there must be a link to create an event which will direct the user to an event creation page.
- The event creation page must have different fields that must be filled in by the user.

- When the user clicks on “Create Event”, there must be a field stating “Type of event” with a drop-down list that must list the 2 options (Course or Seminar) where the user can select the type of event they want.
- There must be a field stating “Title for event” with a text box for input so that the user can type in the title for their event.
- There must be a field stating “address” with a long textbox so that the user can enter in the address of the event.
- There must be a field stating “presenter/guest speaker” with a text box for the user to enter in the name of the presenter/guest speaker.
- There must be a field stating “de-register window time frame” with 3 drop-down lists ranging from “days”, “hours” and “minutes” so that the user can select how long the time frame can be.
- There must be a field stating “Early Bird” with a text box so that convenor can enter a number for the number of early bird days.
- When the user has filled in all fields, there must be a button at the bottom displayed as “Create Event” so that the user can click and post the event to the website.
- If there are fields missing when the user clicks “create event”, there must be an error message stating all the fields that have been missed at the top of the page. E.g. “Please fill in the title.”
- When the user clicks on “create event” with all fields correctly filled in, there must be a message stating “You have successfully created your event” and the user will be redirected to their homepage.
- A confirmation email message must be sent out to the user after creating their event with the relevant information for their event.

Estimate: 5 Story points

Priority: 9

US16 - As a UNSW staff, I should be able to post the dates for my event so that people know which dates to attend.

Acceptance Criteria:

- When the event to be created is a seminar, the website must allow the user to enter multiple dates by having a field for “number of dates” with a drop-down list of numbers. The number chosen will generate multiple drop down lists for the user to input the date.
- e.g. Number of dates: 2
 Day 1: day_ month_ year____
 Day 2: day_ month_ year____
- When the event to be created is a course, the website must allow the user to enter a date in the ‘enter date’ field with drop down lists for day, month and year for the event to run across.

Estimate: 0.5 Story points

Priority: 7

US17 - As a UNSW staff, I should be able to write a description for my event so that people know what my event is about.

Acceptance Criteria:

- When the staff user clicks on “create event”, there must be a field stating “description” with a long and wide textbox for the user to type their description of the event. The textbox must be scrollable so that the user can enter in a long description.

Estimate: 0.5 Story points

Priority: 7

US18 - As a UNSW staff, I should be able to limit the number of attendees for my event to avoid overfilling the hosted area.

Acceptance Criteria:

- When the staff user clicks on “create event”, there must be a field stating “Max amount of attendees” with a small text box so that the user can input the limit for the event.

Estimate: 0.5 Story points

Priority: 7

US19 - As a UNSW staff, I should be able to view all the attendees that have registered for my event so I know who is attending.

Acceptance Criteria:

- When the user is a staff and is also the convenor, the event page should update with all the information about the event and also extra features for the convenor such as a list of all attendees at the bottom of the page.

Estimate: 0.5 Story points

Priority: 7

US20 - As a UNSW staff, I should be able to upload a picture relating to my event so that I can attract people about my event.

Acceptance Criteria:

- When the staff user clicks on “create event”, there must be a field stating “upload image” with a clickable button that can allow users to upload an image from their computer to show on the event page. There must be a file size limit clearly displayed next to this field.
- If the image is too large, there must be a message stating “image file size is too large”
- If the image is within the size limit, the image name should show as being uploaded. E.g. image1.jpg uploaded

Estimate: 2 Story points

Priority: 7

US21 - As a UNSW staff, I should be able to close the event after it has finished to inform users that the event has past.

Acceptance Criteria:

- When the user is a staff and is also the convenor, the event page should update with all the information about the event and also extra features for the convenor such as a button to close the event to indicate the the event has finished.
- After the user has clicked on the “Close Event” button, a message must be displayed stating “You have successfully closed the event”.

Estimate: 2 Story points

Priority: 7

US22 - As a UNSW staff, I should be able to cancel the event if I no longer wish to continue hosting the event.

Acceptance Criteria:

- When the user is a staff and is also the convenor, the event page should update with all the information about the event and also extra features for the convenor such as a button to cancel the event if the user no longer wants to host the event.
- After the user has clicked on the “Cancel Event” button, a message must be displayed stating “You have successfully cancelled the event”.

Estimate: 2 Story points

Priority: 8

ES7 – As a guest user, I should be able to have my own account so that I can easily manage all events that I am involved in.

US23 – As a guest user, I should be able to register for an account on the login page so that I can view and attend events that I am interested in.

Acceptance Criteria:

- When the user is a non-UNSW member, the login page should have an additional option for guests to register with name, email and password.
- The system should check if email has already been allocated and if not, the authentication details should be stored and a “successful” message should appear after guest has clicked submit.
- After creating an account, the guest should be able to login with their authentication details and should be directed to a dashboard where they can see all events that they are attending or speaking.
- If the guest is also registering as a speaker, there should be a description box for the guest to fill out about themselves since the speaker will have their own profile linked to their speaking event.

Estimate: 4 Story points

Priority: 9

US24 – As a guest user, I should be able to attend an event that I am interested in by clicking on the event and pressing ‘attend’ if there are enough spots so that I can reserve my attendance for that event.

Acceptance Criteria:

- When the user is a guest and is on the event page, the system should identify that the user is a guest and should notify the user of the guest fee after the user clicks on attend.
- The event page should also notify of an early bird period for the event so that guests can receive a 50% off discount on the guest fee.
- A successful message should appear after the customer pays and their attendance should be booked for the event.

Estimate: 2 Story points

Priority: 6

US25 – As a guest user, I should be able to pay for attending fee so that my name will be on the attendance list for the event.

Acceptance Criteria:

- When the user is a guest and the user clicks on reserve button when viewing the event page, a pop up message will appear asking “Are you sure you would like to attend event? You will be charged \$x amount”.
- When user is a guest and is also within the early bird period, the fee is discounted to 50% off and a message will be displayed when they press reserve.
- When guest user presses “confirm”, a confirmation email is sent to them about the event and their reservation.

Estimate: 3 Story points

Priority: 5

US25 – As a guest user, I should be able to deregister my attendance within the deregister period if I am not interested anymore.

Acceptance Criteria:

- When user is guest and is not interested in event and wants to deregister, a deregister button will appear only if it is within the deregister period.
- If user clicks on deregister button, they are removed from attendance list and an email will be sent out confirming they have cancelled their reservation.

Estimate: 3 Story points

Priority: 5

US26 – As a guest speaker, I should have my profile description linked to the event page so that people viewing can click on my name and read about me.

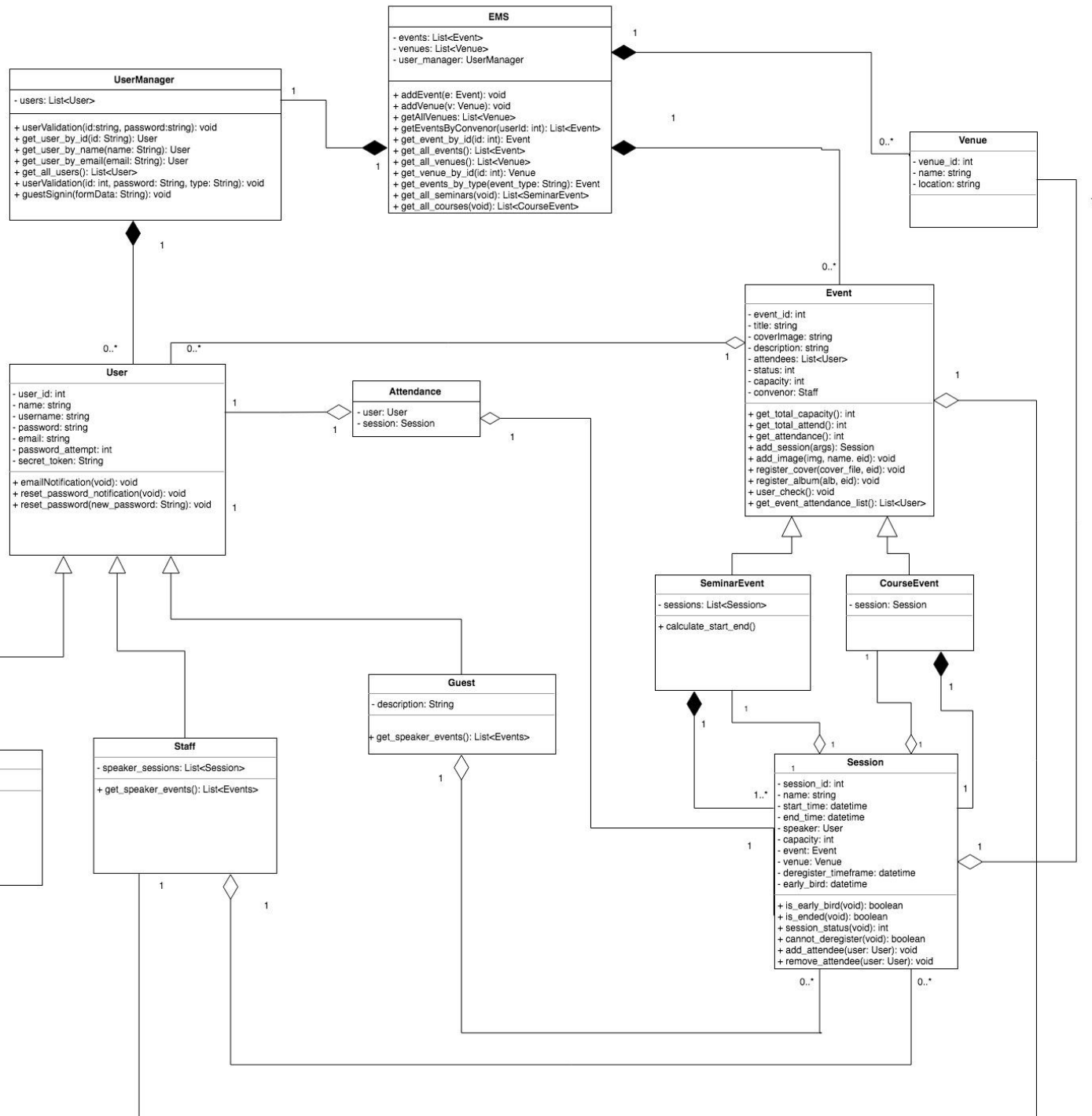
Acceptance Criteria:

- When user is a guest speaker, a guest profile can be found when accessing the event page and clicking on the guest speakers name.

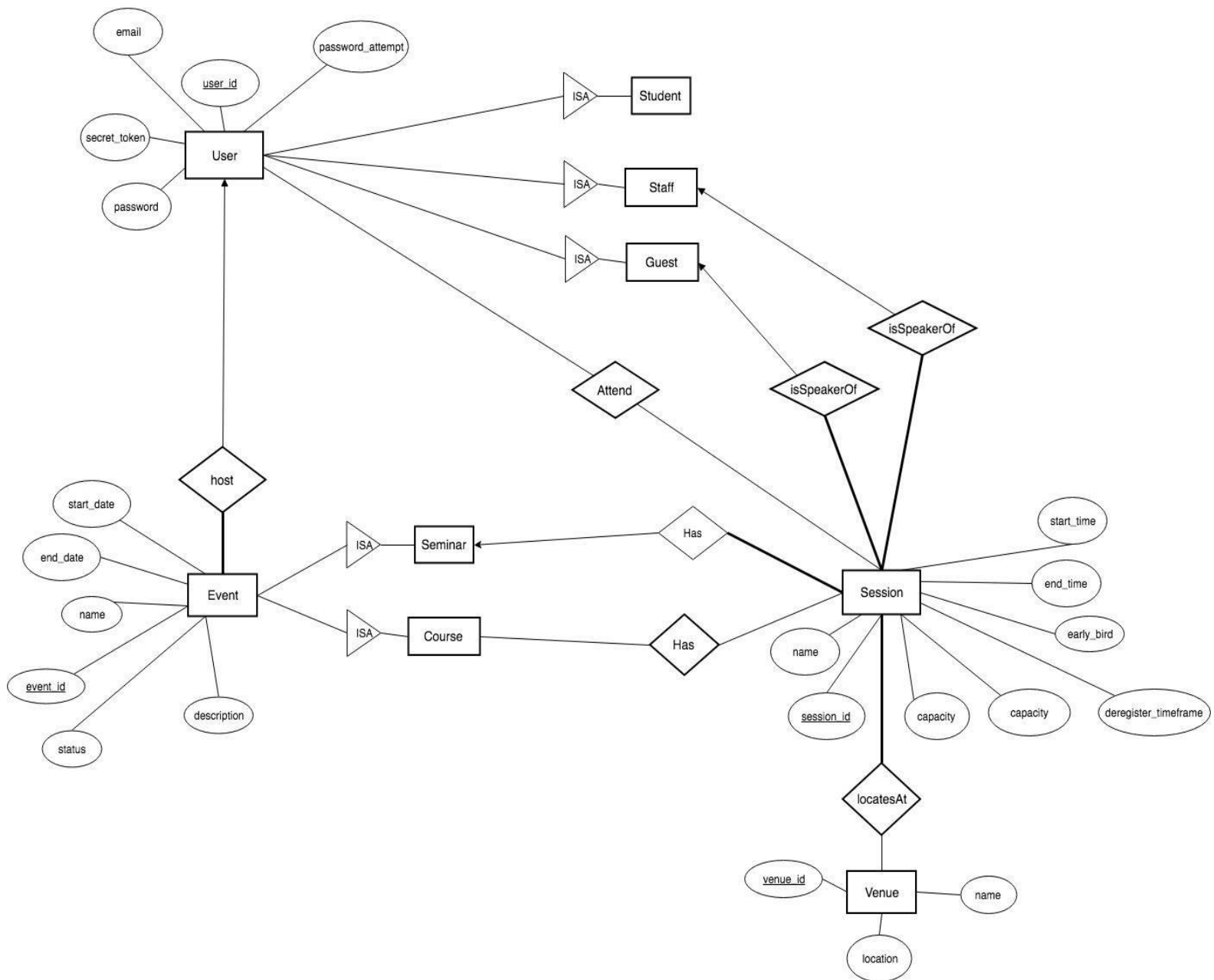
Estimate: 4 Story points

Priority: 4

Class Diagram



Entity Relationship Design



LOGBOOK

We organized stand-up meeting every week before the tutorial to check the current process and obstacles that we have from tasks. And raised a new plan and split tasks for each one for following week.

1	Date: 09-04-2018
Outcomes	
<ul style="list-style-type: none"> - Created GitHub repo <li style="padding-left: 20px;">- Discussed task assignments <p>Austin:</p> <ol style="list-style-type: none"> 1. <i>Epic Stories</i> 2. <i>Acceptance Criteria</i> <p>Henry:</p> <ol style="list-style-type: none"> 1. <i>User Stories</i> <p>Chen:</p> <ol style="list-style-type: none"> 1. <i>UML Class Diagrams</i> <ul style="list-style-type: none"> - Work on epic stories first (early in week) so that there is time for user stories, acceptance criteria and class diagram - Since User Stories component is large, Austin will help out Henry with parts such as acceptance criteria if Henry needs help - Based on Epic and user stories, Chen will make class diagram 	
Reflection	
<p>Epic Stories component was not too hard since it was high level abstraction. Used RGB model to create each epic story from the problem statement given.</p> <p>User stories were then just broken down from the epic stories and many cases were considered from that high abstractions story.</p>	
Plan	
<p>Henry: set up database</p> <p>Austin: implemented classed base on UML</p> <p>Chen: implemented classed base on UML</p>	
Obstacles	
<p>Since we have only been given a week before the first iteration, there has been some time constraints due to other assignments from other subjects and also from work outside of university.</p>	

2	Date: 16-04-2018
Outcomes	
<ul style="list-style-type: none"> - Discussed task assignments <p>Austin:</p> <ol style="list-style-type: none"> 1. <i>Create classes based on UML diagram</i> <p>Henry:</p>	

<ol style="list-style-type: none"> 1. <i>Create/manage database (SQLite)</i> 2. <i>refactor whole structure</i> <p>Chen:</p> <ol style="list-style-type: none"> 1. <i>Create classes based on UML diagram</i> <p>- Discussed how we will store all necessary information i.e. accounts, events etc</p> <p>- Since Henry has experience with databases, primarily SQLite, he will write code that will help us with our website. Austin and Chen will then make the skeleton of all the classes that need to be implemented. We gave ourselves a week to finish the first draft of our back end product.</p>
Plan
<p>Implementation basing on design</p> <p>Austin: Dashboard</p> <p>Henry: user management, index</p> <p>Chen: add event page</p>
Reflection
Iteration 1 was successful and we were all happy with our outcome.
Obstacles
We will all need to be on the same level in terms of knowing how to utilise the database if we are going to write methods that query the information needed.

3	Date: 20-04-2018
Outcomes	
<p>- Discussed about how the website will look like and which front end programming languages to use ie. JavaScript, Split work into manageable portions.</p> <p>Austin:</p> <ol style="list-style-type: none"> 1. <i>Create the dashboard</i> 2. <i>Make dashboard similar to the index page</i> <p>Henry:</p> <ol style="list-style-type: none"> 1. <i>Manage database, create index page</i> 2. <i>user management(login/logout), navbar</i> 3. <i>reusable component of list events</i> <p>Chen:</p> <ol style="list-style-type: none"> 1. <i>Create add event page.</i> <p>- Will give ourselves a week to finish our portion and will meet back on Monday to discuss our next plan of action.</p> <p>- Each team member is responsible of styling their portion using front end techniques but will need to make website look consistent in its looks and feel.</p>	
Plan	
As we have basic working prototype, we will continuous implementation to accomplish all requirements.	

Austin: reset password page, styling dashboard Henry: set up email notification, create event detail page Chen: continue working on add event page
Reflection
Creating classes and also the methods for those classes was a success. We did not need to change our diagrams while creating our model.
Obstacles
We've encountered "python import loop" issue, solved by splitting flask_sqlalchemy into a separated file.

4	Date: 23-04-2018
Outcomes	
<p>- Continued working on our portions. Split up more work between the three of us.</p> <p>Austin:</p> <ol style="list-style-type: none"> 1. <i>Continue styling dashboard. Start on reset password page.</i> <p>Henry:</p> <ol style="list-style-type: none"> 1. <i>Manage database</i> 2. <i>set up email</i> 3. <i>created event details page and implemented reserve/cancel reservation for session</i> <p>Chen:</p> <ol style="list-style-type: none"> 1. <i>Create add event page.</i> <p>- Will give ourselves a week to finish our portion and will meet back on Monday to discuss our next plan of action.</p> <p>- Will also need to start making tests for each component of website so that nothing breaks. Each team member will need to test their own portion.</p>	
Plan	
Process is on the track so far as our project stratifying all basic requirements, we will continue to work on own part to improve UI, security, error handling.	
Reflection	
Website is starting to work together with all its components to show the correct data. Was able to also style the website so that it looks good.	
Obstacles	
None at this point.	

5	Date: 06-05-2018
Outcomes	
<p>- Continued working on our portions. Split up more work between the three of us.</p> <p>Austin:</p> <ol style="list-style-type: none"> 1. <i>Start on reset password notification page.</i> 2. <i>Continue styling dashboard and reset pages.</i> 	

Henry:	
<ol style="list-style-type: none"> 1. <i>Managed maximum login attempt</i> 2. <i>added filter to list_events component and it can be used in dashboard page</i> 3. <i>improve security (avoid url bypass)</i> 4. <i>add notification for logout/reserve event</i> 	
Chen:	
<ol style="list-style-type: none"> 1. <i>Continue adding fields for add event page and also album</i> 	
Reflection	
Identified some bugs in front end but was able to fix it through communicating with team mates about these bugs. Most user stories have now been implemented and we now need to create tests to see if anything breaks. Also refined code so that easy to read and maintain.	
Plan	
The last week before completed iteration 1, we decided to refactor some of our code to improve portability, allowing project to be more maintainable.	
Obstacles	
Hard to wrap reusable component in flask	
6	Date: 13-05-2018
Outcomes	
- Continued working on our portions. Split up more work between the three of us.	
Austin:	
<ol style="list-style-type: none"> 1. <i>Style Dashboard page</i> 2. <i>refactored methods of pagination</i> 	
Henry:	
<ol style="list-style-type: none"> 1. <i>Created deregister time frame for back-end support</i> 2. <i>UI improved</i> 	
Chen:	
<ol style="list-style-type: none"> 1. <i>Added deregister timeframe filed on add event page</i> 2. <i>add event page validation</i> 	
Reflection	
Ensured all functionalities have been implemented according the user story that we wrote, improved readability of the code by code refactoring and cleaning up	
Plan	
Make plan and split tasks for each one for iteration.	
Austin: profile page	
Henry: guest sign in, registration fee	
Chen: improve add event	
Obstacles	
Manual test for each functionality seems to be vulnerable and time consuming as we haven't set up unit test and integration test yet.	

7	Date: 21-05-2018
Outcomes	
<p>- Continued working on our portions. Split up more work between the three of us.</p> <p>Austin:</p> <ol style="list-style-type: none"> 1. <i>profile page for guest speaker</i> 2. <i>refactor code for event management</i> <p>Henry:</p> <ol style="list-style-type: none"> 1. <i>implemented guest login</i> 2. <i>improved UI profile page</i> 3. <i>added attendance bar on list event component</i> 4. <i>added guest registration modal</i> 5. <i>implemented registration fee</i> <p>Chen:</p> <ol style="list-style-type: none"> 1. <i>fixed bugs for add event</i> 	
Reflection	
<p>Iteration 2 kicked off and continued task splitting. We essentially implemented all required features (guest login/sign in, profile page, attendance indicator on index page) by specification as the most complicated part (user is able to attend multiple session for a event) was implemented in last iteration. Also we fixed up some remaining bugs from last iteration.</p>	
Plan	
<p>Refactor some codes on our project, complete unit testing and try to find issues, enhance error handling.</p>	
Obstacles	
<p>Our reusable component of list event doesn't work on profile page (URL of image used relative path of current page rather than abs path from root), however, it can work smoothly on dashboard page. Solved by adding a url_for with a function to retrieve image from backend.</p>	

8	Date: 28-05-2018
Outcomes	
<p>- Continued working on our portions. Split up more work between the three of us.</p> <p>Austin:</p> <ol style="list-style-type: none"> 1. <i>added exception handler</i> 2. <i>added test cases</i> 3. <i>drafted final report</i> 4. <i>updated velocity chart</i> 5. <i>update user stories</i> 6. <i>adjusted reset password/forget password page and allows it work for guest user</i> 7. <i>fixed bugs for forgot my password and guests</i> 8. <i>fixed display messages</i> <p>Henry:</p> <ol style="list-style-type: none"> 1. <i>fixed bugs for add event</i> 	

2. *code clean up*
3. *completed ER diagram basing on our DB design*
4. *adjusted UML diagram*
5. *added more test cases*
6. *added more exception handler*
7. *implemented early bird*
8. *adjusted models with constructor*

Chen:

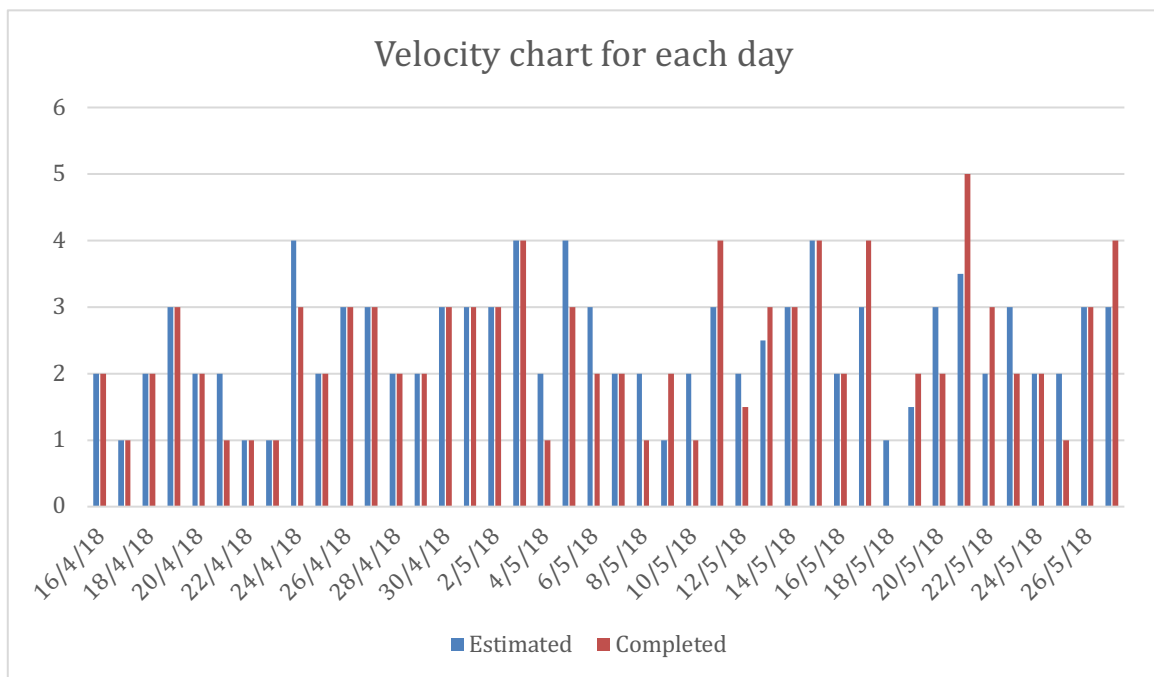
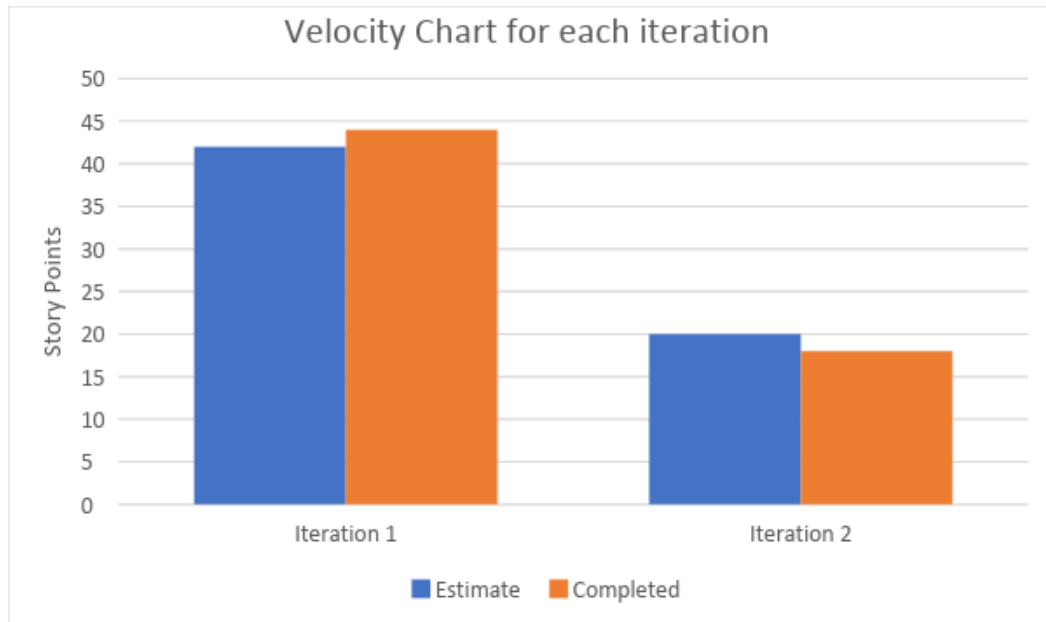
1. *fixed bugs for add event (typeahead duplication, empty input when switch)*
2. *fixed wording*
3. *updated message dynamically when event listing has nothing*

Reflection

We focused on testing and enhancing after completion of all features before the final iteration of development. Some bugs are identified from unit and manual testing for different functionalities as we don't have integration test. According to these issues identified, we fixed the bugs as well as implemented some exceptions handlers and error handling, especially for edge case to improve scalability and reduce issue identifying process of our application.

Obstacles

None at this point.



Authentication Details

Staff

Username: 1234

Password: 1234

Student

Username: 1111

Password: 1111

Guest Speaker

Username: hi@gmail.com

Password: 1234

Reflection

Overall, the project was very enjoyable for all of us in the team. We were able to finish each component very early within the given iterations which left us a lot of time to debug. Task splitting could have been better so that each of us could have contributed equally but was fair since we had our own obstacles to take care of. If we had to do it again, we would make sure that we meet up more so that more decisions could be made and would also work together at the same time so that we can help each other in real life rather than online. We took a long while on UML and structure design, as well as extra time on refactoring and re-implementation since we didn't sit together to complete the user stories writing such that it exists misunderstanding when we were trying to implement. Instead, we should simulate the real scenario of requirement gather, analysis and specification between business domain owners, customers and developers to achieve mutual agreement and completely understanding on requirements.