Individual Task Assignment for the iteration 1:

Task Description:

1. Create the models: Create the model classes Event, Keyword and Location.
2. Feed the dummy data for the model classes: Create the seed data for these model classes.
3. Specify the validation for each model classes: An event must have title, description, date and time. The date and the time of any event must in a proper format i.e. that must be a valid date and valid time. Both image and link attributes can be empty. Likewise, for the model class Location, the attribute room can be empty but remaining attributes must be present. Similarly, tag must be present in the Keyword model. These validations are to be implemented
4. Create the association between Event-Location and Event-User
   The association between the event class and the location class should be created. Similarly, create the association between Event-User model classes.
5. Design the structural component for Detailed Event Page, Home Page and Edit Event Page

   The mentioned pages include the basic design to ensure the understanding of the user-interface by the customer.
6. Features: Get the list of the events at the home page, Edit the event

   The default home page must present the list of all events. Editing feature should be implemented. The owner can only edit any valid event.
   [Note: as the designing the registration page was under Qiong's task assignment so the feature of user account registration was assigned to her.]

Task Evaluation: The instructor can use following measures to ensure the quality of the task:

- Whether the attributes of a model class follow the validation specified or not.
- Whether the test cases are properly generated or not
- Whether the associations between the model classes are in accordance with the class-diagram under "misc" folder in github.
- If the home page shows the list of all the stored events or not.
- If the changes made for the events from the User interface are reflected at the database or not.

Outcome of the mentioned task:

The models creation, association, validation has been successfully implemented. The seeded data for these models can be viewed from the user interface as well as database. Test cases for the model validations are created. These test cases can be run successfully.

The designed pages only contain the basic functional components with minor styling just to ensure user-friendliness. The list of all the events stored in the Event model can be achieved in home page. The user can only edit the event she/he created.

Unplanned tasks:

Added validation Event model:
event_date: user should not be allowed to enter the time prior to the current date
event_time: user should not be allowed to enter the time (if the date is same)that is prior to the current time
link: added format in the link attribute in order to ensure that the entries for the link reflect the url format.

Note: as adding allow_blank validation to the image attribute required at least one another validation, which was not planned, that validation is not added here. As the field can be empty, so no validation was added for that attribute.

How to evaluate this task:
The test cases were generated to ensure these validations. The successful test case run ensures their successful implementation.

Outcome of the task:
The user is not able to enter the event with the date prior to the current date. The user is also not able to enter the time, which is prior to the current time. While providing the seed data, the valid time and the valid date are to be provided to ensure successful test case run.

Implemented a feature for marking an event as "Invalid":
In order to fully implement the editing functionality of a particular event by the creator of the event, the user should be able to Mark it as "Invalid". Even though, this feature was not assigned for this iteration, it was implemented because of that reason. However, once any event is marked as invalid, it can't be marked as valid.

How to evaluate:
If the creator of the event is able to mark any event s/he created as invalid and the changes are reflected in database as well as in the user interface, then the task is successfully accomplished.

Outcome of this task:
The user who created the event can only edit the event i.e. s/he can only mark it as invalid. If the event is marked as invalid, it can't be marked as valid again. As marking the event as invalid by the creator of the event refers as the event was cancelled. Hence, the cancelled event can't be again happening.