Iteration 1 Documentation

Team:

Gangs of Aggiesthan

Teammates roles:

Product Owner: Utkarsh Chanchlani

Scrum Masters:

Iteration 0	Ankit Garg
Iteration 1	Ranjith Tamil Selvan
Iteration 2	Harish Chigurupati
Iteration 3	Aswin Periyadan Kadinjapali
Iteration 4	Jatin Kamnani

Customer meeting details:

Date: 03/27/2019
Time: 5.30 p.m.
Place: Rudder Tower, Room No.1005

Summary:

During the meeting, we presented the working prototype of the login/register user interface of the application deployed on Heroku. The user interface as designed in Iteration 0 was discussed and agreed upon. We also went over the current design and discussed the priority stories for the coming iterations.

Additional inputs from customer:

1. The customer asked for inclusion of logo on the home-page.
2. Customer requested for additional feature to give SI leader provision to edit attendance per student for past sessions.
3. Customer also requested for additional feature to give SI leader provision to create a batch of similar sessions, in addition to creating sessions individually.

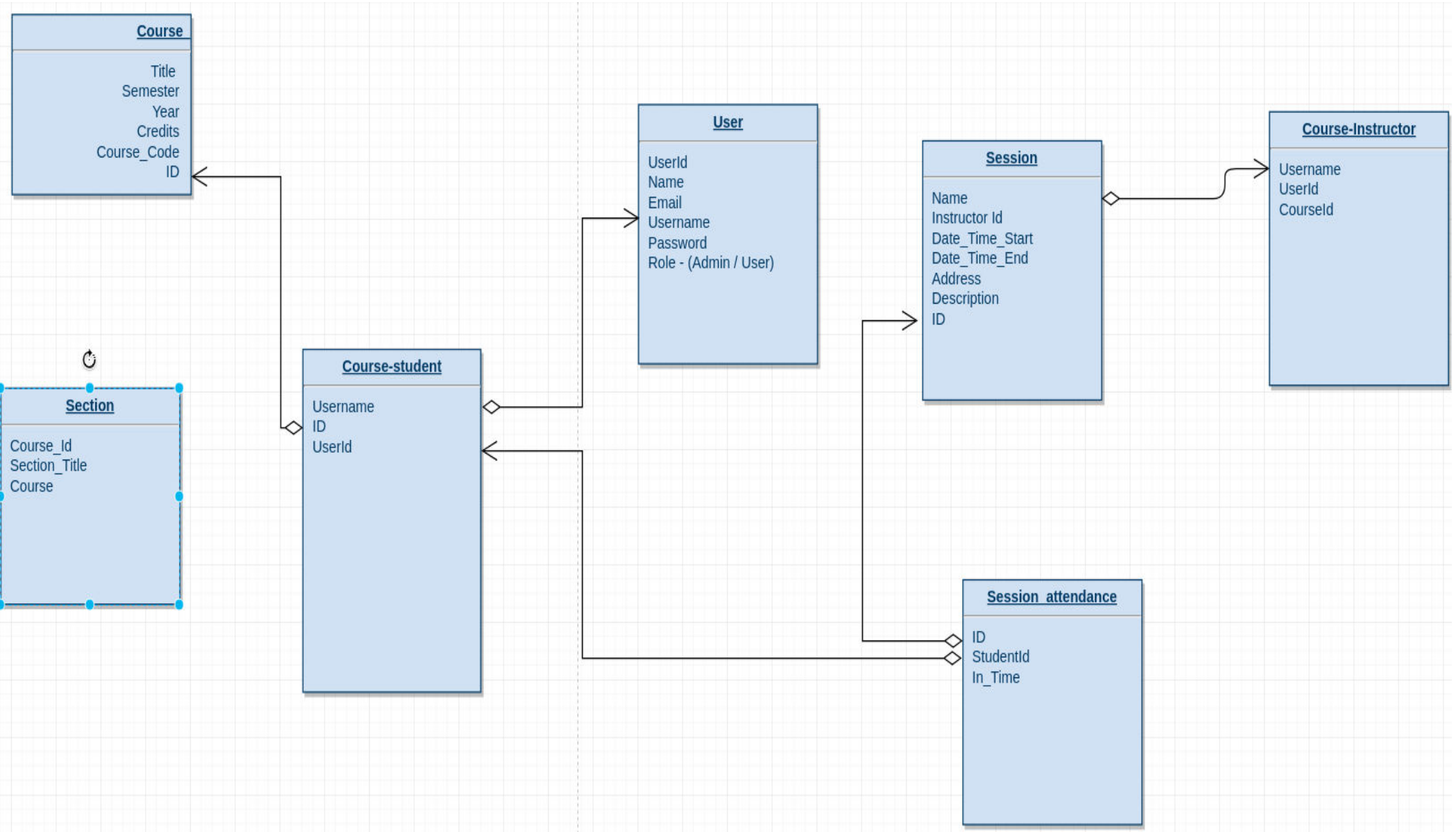
User Stories Implemented in Iteration 2:

Feature 2: User (Student/Supplemental Instruction (SI) leader) should be able to see the dashboard after login/signup process.

Sub stories:

1. Design Front end for the dashboard page. (3 points) *Implemented*
   a. Dashboard should display 3 buttons on the top- SI leader, Student, and logout button (this button should redirect to the home page/login page).
   b. Under SI leader/student (common to both) page, we must see the list of courses available and when we click each course, following fields should be displayed.
      - Course ID
      - Course Code
      - Title
      - Semester
      - Year
      - Credits
      - Sessions.
2. Implement and design the course GET calls. (2 points) *Implemented*
3. Implement and design Admin's API for course registration, SI leader enrollment, and Student enrollment. (3 points) *Implemented*
4. Integrate the dashboard page with the API calls. (1 point) *Implemented*
5. Design UI for the admin page. (1 point) *Implemented*
   a. The page should include all the CRUD operations required for the admin.

Design UML Diagram



The Design diagram shows the relationships across the entities of the system, namely- User, Session, Course-student, Section, Course, Session attendance, Course-instructor. The attributes per entity are displayed in the UML diagram.

Each student/SI leader will be associated with a User object with role -user. Course-student is an object per student per course. This object is created by admin for valid student users, i.e., who have registered for that course. The course-student object will be associated with the User (student) once the student has registered with the SI system. As such, Course-student has two foreign-keys, namely Username and UserId, which are primary keys to user and course objects respectively. Each course can have multiple sections. An SI leader can create a session, which will be associated with a session object. As such, a session object will have a foreign key 'InstructorId' to the Course-instructor. Session attendance will be created per session, and will have foreign keys to session and course-student objects.

Pivotal Tracker:

https://www.pivotaltracker.com/n/projects/2318414

Github:

https://github.com/tamu-asc/ascss
Github Tag for Iteration 1: https://github.com/tamu-asc/ascss/tree/Iteration2

Heroku:

https://tamu-ascss.herokuapp.com/

Auto-deployment:

We deploy the application automatically from the master branch. Master branch is our production branch and any code merged into that will be automatically deployed into the heroku production application. Manual deployments can be triggered via heroku console as well. Master branch is protected which means no one can commit directly into the master branch.

Grading Approach:

Since we are doing a non-legacy project and setting up the platform from scratch, and the user requirements are high, we are concentrating more on the behavior of the features. Since we focused more on the setup and implementation of the first-level features, we have not followed Test Driven Development.