

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 | Jatin Kamnani |
| Iteration 3 | Harish Chigurupati |
| Iteration 4 | Aswin Periyadan Kadinjapali |

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.

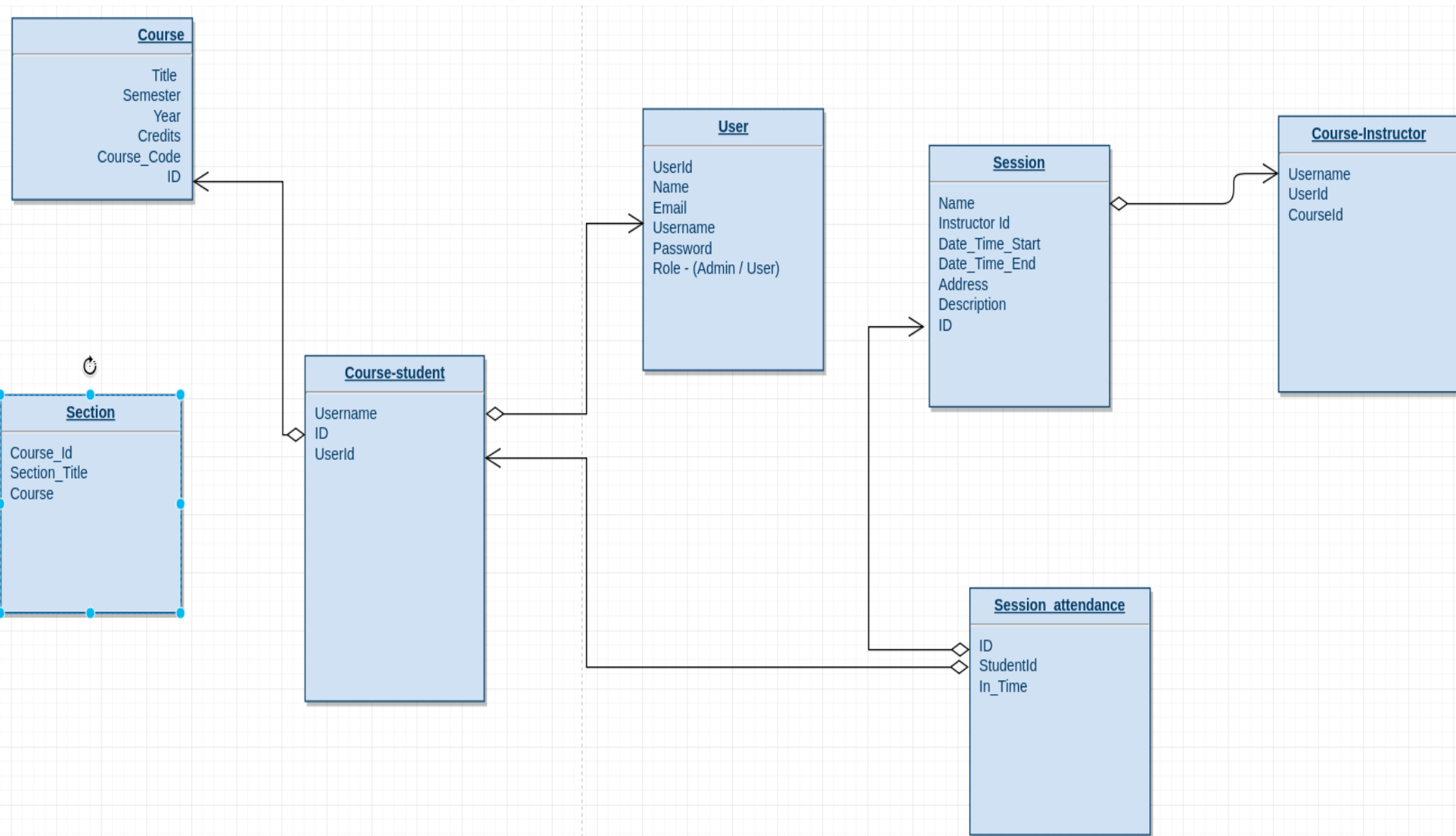
User Stories Implemented in Iteration 1:

Feature1: User (Student/Supplemental Instruction (SI) Leader) should be able to register/login to the Supplemental Instruction system at application startup.

Sub Stories:

1. Environment setup (3 points) *Implemented*
2. Design front-end home-page (2 points) *Implemented*
 - a. Home-page should display login page (for user – student/SI Leader). email-address and password would serve as login-credentials for a user.
 - b. Provide register link 'Create a new account' in home-page for users to register with SI system, if account is not already registered.
3. On clicking 'Create a new account', user should be redirected to 'register page' (2 points) *Implemented*
 - a. Register page displays form with the following entries:
 - i. First Name
 - ii. Last Name
 - iii. Username
 - iv. Email address
 - v. Password
 - vi. Password Confirmation
 - b. Provide register button to complete and submit form. Also, provide link 'Already have an account?' that should redirect the user to login on home page.
4. Design Business logic for Login/Register (2 points) *Implemented*

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/Iteration1>

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.