Team Project

Start Assignment

Due May 12 by 11:59pm **Points** 100 **Submitting** a website url or a file upload

Use Case:

Implement an end2end **HealthClub** Membership Management system for your favorite Health club/Gym,

The emphasis here is on team collaboration, so the points awarded will be based on individual contributions to the team and how the team performed overall.

- **Components**
  - **APIs -** input and output of API should be in JSON and should include error handling and validation of inputs
  - APIs will be demonstrated using a Web/mobile UI
  - UI is accessed by Members, Non-Members, and HealthClub employees (who are admins) - (3 roles)
  - APIs should support following functionality:
    - View Home page showing information about the Gym and memberships available and Class schedules - viewable by all users
    - Enrolled and logged in Members:
      - View members page - showing your individual class schedule,
      - View Activities in the past week/month/last 90 days
      - Signup for classes in advance
      - Log hours on Treadmill, Cycling, Stair machines or weight training
    - Healthclub employees :
      - Enroll new members
      - Checkin members into the Gym each day
      - Signup non-members for free trials
      - View analytics dashboard showing User activity summarized by location
        - Classes and enrollment by day/week

        - Hours spent in the gym by day/week/month
        - Number of visitors by the hour each day, weekday, weekend
        - Any other useful dashboard (your team can get creative!) that will help planning gym hours, schedules, equipment inventory
    - APIs and UI functionality will be available based on Roles specified above
    - Assume the gym membership will be valid in multiple locations and Home page will let you select the Location to view corresponding schedules

- - Deploy API to **AWS in an Auto Scaled EC2 Cluster** with Load Balancer (or another cloud provider)
  - Develop a Web or mobile UI that will make use of the APIs
  - Create your own database with mock data for classes, locations, schedules, instructors

Requirements:

- Each team member must ***own at least one of the component***s in the Team project.
- Keep a **Project Journal on GitHub** to include:
  - **Weekly Scrum Report** (i.e. weekly version of daily scrum) which answers the three daily stand-up questions:
    - What tasks did I work on / complete?
    - What am I planning to work on next?
    - What tasks are blocked waiting on another team member?
  - Select two of the **XP Core Values** and keep a journal of how the team kept these values throughout the project. Report this in your Project Journal with the weekly Scrum Report submissions:
    - Communication
    - Simplicity
    - Feedback
    - Courage
    - Respect
- **Maintain Weekly Scrum Task Board (in GitHub as a Project Board or in a Google sheet)**
  - Update the **Story** on your Task Board
  - Keep track of **remaining effort** and progress on a Team Task Board.
  - Use (and modify) the **Google Task Sheet** Template at:

    - Click on this **LINK** (https://docs.google.com/spreadsheets/d/1RBzwUDx9QG7Uy8ayiFBBuhWBaJCrK5dV5T9eN2ZEfp8/edit?usp=sharing) . (Make adjustments to fit your team size)
    - Track your Team's **Burndown Chart** in this Sheet.

- Maintain the project artifacts and code in an assigned **Team GitHub Repo (you will get a Google classroom invite to create a private repo - one per team)**
- Create **UI Wireframes**
  - Create UI wireframes for each of the screen in your team's solution
  - (this can be done by hand or electronically with a tool like "Pencil")
- Create an Overall **Architecture Diagram** showing:
  - Software Components and their Public Interfaces
  - The Dependencies between Components
  - The Relative Relationship of how these components are Deployed
  - Recommendation: Use UML Deployment/Component Diagram Notation.
    - **http://agilemodeling.com/artifacts/deploymentDiagram.htm** (http://agilemodeling.com/artifacts/deploymentDiagram.htm)
    - **http://agilemodeling.com/artifacts/componentDiagram.htm** (http://agilemodeling.com/artifacts/componentDiagram.htm)
- Maintain a README markdown file in the Team's GitHub Repo.
  - Include all **Diagrams**, **Design decisions** and the overall **Feature Set** of the project
- **Project Demo**
  - Give a demo of your teams working prototype on "Demo Day"

Grading:

- Teams will be be graded with a **Team Score during Demo Day.**
  - **100 points**
- Individual deductions will be made to the Team Score based on contributions to be judged by:
  - **Completeness and Functioning Demo of your Component (as noted on Demo Day)**
  - **Frequency and Quality of commits** to the project Github.
    - As such, it is expected that all contributions must be visible via Github. See the following guidelines for how GitHub counts contributions: **https://help.github.com/articles/why-are-my-contributions-not-showing-up-on-my-profile/** (https://help.github.com/articles/why-are-my-contributions-not-showing-up-on-my-profile/)

- **Rubric:**

- **Architecture/Design: 10%**
- **Implementation of requirements (working software): 70%**
- **Agile Scrum Process (includes Weekly commits and submitting Sprint artifacts, XP values): 20%**
- **Github insights - expectation is that every member has similar amount of contributions to code-lesser contributions will result in individual deductions**

Submission (One per Team): -include this information in the Readme section of the repo:

- **Your Team Name**
- **The names of each team member**
- **A summary of areas of contributions (for each team member)**
- **Link to your team's GitHub Repo**
- **Link to your team's Project Board (on GitHub)**
- **Link to your team's Project Journal (on GitHub)**
- **Link to your team's Google Sprint Task Sheet**

Example Format for Weekly Stand-up (i.e. Daily Scrum) and Final Burn-down Chart & Task Board:

Daily Scrum + Burndown Chart

Team Name, Sprint #1

Team Member Name

John Smith

What I did since the last daily scrum:

- Draw UML Class Diagram (done)
- Draw Sequence Diagram (not done, est. 2 more hours)

What I plan to do today:

- Draw Sequence Diagram
- Write Unit Tests

What blockers I have:

- I am waiting on the interface definition for my FooBar class. We need to define this ASAP.

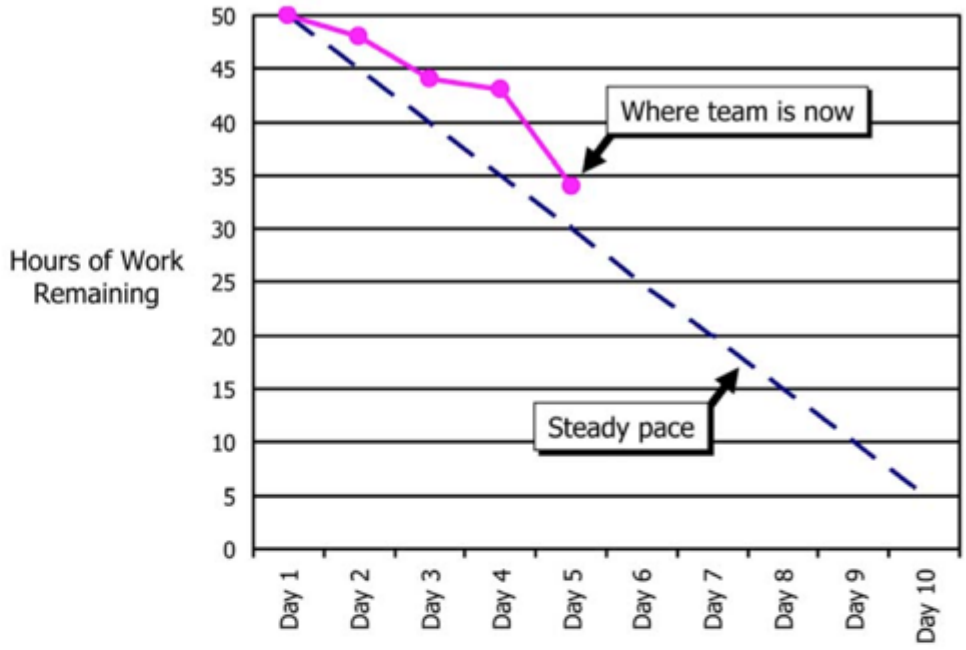

Figure 6. Burndown Chart

				Hours of Work Remaining on Each Day of the Sprint									
Backlog Item	**Task**	**Task Owner**	**Initial Estimate**	**Day 1**	**Day 2**	**Day 3**	**Day 4**	**Day 5**	**Day 6**	**Day 7**	**Day 8**	**Day 9**	**Day 10**
Enable all users to place book in shopping cart	Design business logic	Sanjay	4	4	3	3	1	0					
	Design user interface	Jing	2	2	1	1	1	1					
	Implement back-end code	Philip	6	6	2	5	2	0					
	Implement front-end code	Tracy	4	4	3	2	2	2					
	Complete unit testing	Sarah	4	4	3	3	3	3					
	Complete regression testing	Sarah	2	2	3	3	3	3					
	Write documentation	Sam	3	3	4	2	0	0					
Upgrade transaction	Merge DCP code and complete layer-level tests	Jing	5	5	2	2	1	0					

processing module (must be able to support 500 transactions /sec)	Complete machine order for pRank	Jing	4	4	2	0	0	0					
	Change DCP and reader to use pRank http API	Tracy	3	3	3	2	2	2					
		Total	**50**	**50**	**48**	**44**	**43**	**34**					