COMP 4350 Project Iteration 1

Group H (ListAssist)

Implementation: **(6/11)**

- Fully working features for the user stories planned for iteration 1. **(1/6)**

"As a user, I want to be able to check an item off the list when I no longer need it."- does not work.

"As a user, I want to be able to add an item to one of my lists."- does not work.

(-5)

- Back-end unit tests (dependencies to UI, DB, or other classes should be stubbed/mocked out) **(3/3)**
- At least one integration test per detailed story **(2/2)**

Design: **(8/9)**

- Good programming practices **(2/2)**
- Proper design patterns **(2/3)**

You should call the same business logic code from MVC and API. You should introduce a business logic layer (a class library) where you implement all the business logics (e.g., listing, creating, updating logics for your application).
ListAssist.WebApp/ListAssist.API <--> ListAssist.BusinessLogic <--> ListAssist.Data
Then changing code for business logic will reflect both of your WebApp and API. Hence, the code becomes more reusable and maintainable. You should not directly call database from controllers. (-1)

- Architecture sketch showing high-level components/services and the important relationships between them plus some important classes within components. **(4/4)**

Planning: **(4.5/5)**

- Team's communication and members' contributions **(3/3)**
- Following the plan **(1.5/2)**

Number of user stories for iter1 was too small and is not even completed. In the end, in total you have to implement 10/12 big user stories for the whole project. So, you have to work fast. (-0.5)

Total **(18.5/25)**

Users are lazy. They do not like to provide ID while creating list. If I provide duplicate ID it throws exception. Do not forget to handle all exceptions.
Plan your stories wisely.