

# Individual Homework 2

[New Attempt](#)

- Due Feb 12 by 11:59pm
- Points 100
- Submitting a file upload
- File Types pdf
- Available Jan 13 at 12am - Feb 14 at 11:59pm

## Change Log:

This assignment has been updated to clarify which User Stories to implement, that input validation is required, where to find sample UML documentation you are expected to produce, and other clarifications.

Tasks 4 and 5 have been updated to clarify what user stories you are to implement.

## Introduction

Using the provided foundational code and materials as well as HW1 and/or TP1, each student must design, document, create, and then test a standalone application demonstrating a set of Create, Read, Update, and Delete (CRUD) and input validation operations for each of the following items with sample data appropriate for the Team Project.

- a question class
- an answer class
- a questions class that supports storing all current questions as well as any subset of the questions (e.g., a subset reflecting the results of a search)
- an answers class that supports storing all potential answers to all stored questions and any subset of potential answers to a question (e.g., a subset reflecting the results of a search)

Note: The above-mentioned subsets might be empty, might contain one or more elements, and may be arbitrarily large. There is no fixed upper limit to the number of elements in a list.

You are urged to look at Ed Discussion to see the kind of question messages students sent and the kind of replies students generated. What information is present in the Ed Discussion system that should be part of the Team Project program? Your testing is expected to show realistic data (e.g., use actual Ed Discussion questions and answers).

You are expected to engage with others in the class, directly or via Ed Discussion, to identify potential attributes for the question and answer classes and use the insights you have gained to produce what you believe is required. Do not blindly implement what you believe others are doing, and most definitely, **do not copy** their work!

Document your code and your automated testing in a manner consistent with the provided materials. Remember, the code tells the computer (and humans reading the code) what to do but does not explain why this code is being used or why some other, possibly simpler, solution was not used.

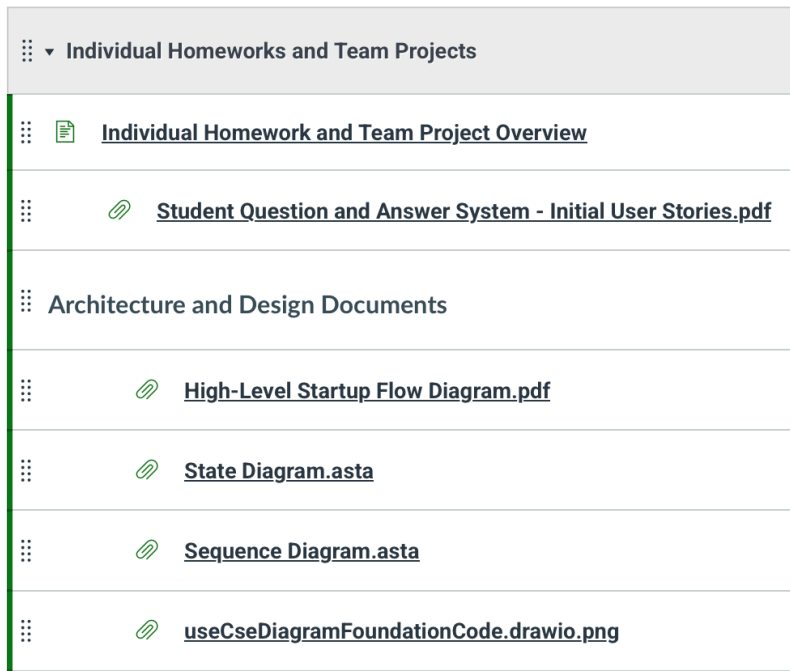
With the requirements satisfied, you must create a short screencast that shows that the application performs each requirement and passes the tests. Use the provided MS Word template, fill it out, and submit it as a PDF where links can be copied and or clicked to access the screencast and your personal GitHub. The detailed requirements for this homework are specified below.

You are encouraged to get help from your classmates if you have issues and offer to help others if you have it running.

## Tasks

This homework requires you to perform the following tasks.

1. Fill in the cover page of the template as required.
2. Create and **document** a subset list of the provided Student User Stories that cover what you believe is required to satisfy all the CRUD and input validation operations based on what you have read (above and elsewhere) and what you have learned from your participation in Ed Discussions and members of the instructional team (e.g., the professors and TAs). You may need to break one or more of these provided User Stories into smaller stories for this assignment as they may be epics. You are **not required** to implement functions beyond CRUD and input validation, but you may choose to implement some that will help you test these requirements.
3. Create and **document** a list of tests that you believe will cover the required functionality (both positive and negative), including input validation and the generation of error messages.
4. Create and **document** a standalone Java application named HW2 that addresses the User Stories from the subset documented in Task 2. (Be sure to produce four UML Class Diagrams and at least one Sequence Diagram using the provided samples in the "Architecture and Design Documents" portion of the "Individual Homeworks and Team Projects" Module as shown below.) Produce a set of test cases from Task 3 that covers the successful execution of your User Stories and detect failures that produce error messages. (Consider using test-driven development.) It is your choice as to whether this is a JavaFX or a console application.



5. Test HW2 to convince yourself that it satisfies the User Stories you created in Task 2 and properly detects improper input with helpful error messages. Produce a screencast that describes the User Stories, show and explain how the test cases cover those user stories, and then demonstrate that the application and the test cases work.
6. Store your application and screencast in your personal GitHub repository. Make sure that only you and your grader have access to your repository.
7. Produce a PDF that contains the documented results from Tasks 1 through 5. Provide links that can be copied and/or clicked to access your HW2 code and your screencast.
8. Submit this PDF before the deadline so there is enough time for the upload of the submission to finish, for Canvas to process the submission, and for Canvas to add it to its data repository before the deadline. Just **starting** the upload before the deadline is **not adequate!** Things can take longer than you think, and networks fail, computers crash, and power goes out. Give yourself time to recover from a surprise.

## Deliverables

A PDF document must be produced that covers the following items. The links in the PDF must be copyable or clickable to access the GitHub copies of the code and screencast.

- Use the following Template for your PDF submission. Fill in this template with the results of the following tasks and submit it. (5%)

**HW2 Assignment Template.docx** (<https://canvas.asu.edu/courses/215181/files/102572828?wrap=1>)

- Task 1: Cover page complete with your name. (5%)
- Task 2: A subset of the Student User Stories. (15%)
  - User Stories cover CRUD operations on the question and the answer classes. (5%)
  - User Stories cover CRUD operations on lists of questions and answers. (5%)
  - User Stores cover both positive and negative outcomes. (5%)
- Task 3: A list of Test Cases. (15%)
  - Testing on CRUD operations on the question and the answer classes using realistic data. (5%)
  - Testing on CRUD operations on lists of questions and answers. (5%)
  - Testing covers both positive and negative outcomes. (5%)
- Task 4: Create and Document HW2 and store it in your personal GitHub Repository. (20%)
  - UML Class and Sequence Diagrams are in alignment with the requirements. (5%)
  - Provide a URL and password to the source code in your GitHub Repository, and a ReadMe provides information to the grader on how to access it, and the source code is consistent with the Class and Sequence Diagrams. (5%)
  - The source code covers all the User Stories. (5%)
  - Source code covers all the Test Cases. (5%)
- Task 5: HW2 Screencast stored in your personal GitHub Repository. (30%)
  - The screencast shows and explains the code and shows that it is in alignment with the documentation. (10%)
  - The code is readable, and the explanations are clear and audible. (10%)
  - The screencast explains how the code accomplishes its purpose. (10%)
- The grader must be able to examine the code in your GitHub, see that it is nicely formatted with internal documentation, and determine that it is consistent with the provided code so most people would assume it had been written by the same author. (10%)