Individual Homework 1

- Due Jan 24 by 11:59pm
- Points 100
- Submitting a file upload
- File Types pdf
- Available Jan 13 at 12am - Jan 26 at 11:59pm

This assignment was locked Jan 26 at 11:59pm.

Introduction

Using the provided foundational code and the provided materials, each student is expected to enhance the code by adding input validation for the various input fields, making several other changes to the code, capturing screenshots (or printing to a PDF) that show that the application performs each requirement, and submitting a properly filled out PDF using a provide MS Word template. The detailed requirements for this homework are specified in the Canvas assignment.

You have been provided the following code elements (three JavaFX programs and a Java Library) to use as the basis for this homework assignment.

First, you have been provided with two small application archives you will need to update HW1 and add them to your Eclipse Workspace.

> **UserNameRecognizerConsoleTestbed.zip (https://canvas.asu.edu/courses/215181/files/100829335?wrap=1) (https://canvas.asu.edu/courses/215181/files/100829335/download?download_frd=1)**
>
> **PasswordEvaluatorTestbed.zip (https://canvas.asu.edu/courses/215181/files/100829271?wrap=1) (https://canvas.asu.edu/courses/215181/files/100829271/download?download_frd=1)**
>
> **Adding Two Applications to Your Workspace.mov (https://mediaplus.asu.edu/embedded?id=5d501aab-0a7f-4fba-8264-b0eed753e4a8&siteId=61e0606e-415d-4001-8206-ffde48430c64)**

Second, you must add the following H2 Database library to your Root directory and make it available to Eclipse. We have provided another screencast to help you do that.

> **H2.zip (https://canvas.asu.edu/courses/215181/files/100859644?wrap=1) (https://canvas.asu.edu/courses/215181/files/100859644/download?download_frd=1)**
>
> **H2 Screencast.mov (https://mediaplus.asu.edu/embedded?id=66f02005-3226-471b-8377-e68bca8d8af9&siteId=61e0606e-415d-4001-8206-ffde48430c64)**

Finally, you have been provided with the FoundationalCode application archive and yet another screencast on adding the FoundationalCode application to your Eclipse Workspace and getting it to run.

FoundationCode.zip (https://canvas.asu.edu/courses/215181/files/100846758?wrap=1) (https://canvas.asu.edu/courses/215181/files/100846758/download?download_frd=1)

FoundationsCode Screencast.mp4 (https://mediaplus.asu.edu/embedded?id=f7d4c746-4956-4f4a-bb46-f82de6829ccb&siteId=61e0606e-415d-4001-8206-ffde48430c64)

(https://canvas.asu.edu/courses/215181/files/100829335?wrap=1)

(https://canvas.asu.edu/courses/215181/files/100846758?wrap=1)

You must have installed the Java JDK 23.0.1, JavaFX SDK 23.0.1, and Eclipse 2024-12 and have verified it by downloading and running the ASUHelloWorldJavaFX application. You are encouraged to get help from your classmates if you have issues and offer to help others if you have it running.

You are allowed (and even encouraged) to work with your Team Project teammates or with other class members to set up your computing environment (Java, JavaFX, and Eclipse) as well as Tasks 1 through 4.

The remaining tasks must be performed by each member of the class without the help of anyone other than a member of the instructional staff (instructors, TAs, Graders, UGTAs).

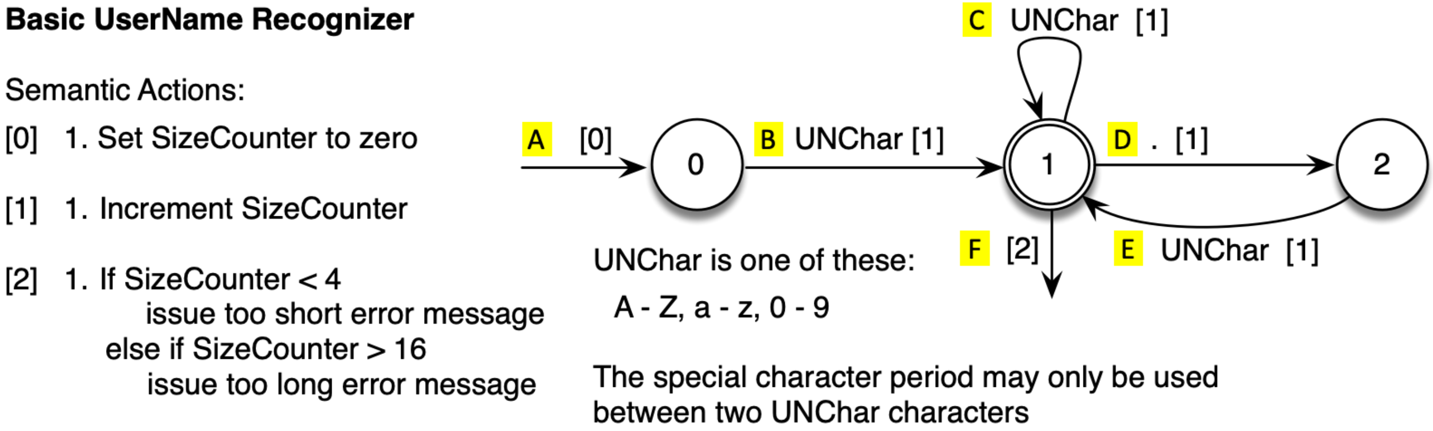
Tasks

This homework requires you to perform the following tasks.

1. Download and verify you can run the Foundational Code on your computer.
2. Identify the input values the Foundational Code requires the user to enter.
3. Study the Finite State Machine (FSM) Input Validation Screencast.

   Input Validation via FSMs - 2025-01-16.mp4 (https://mediaplus.asu.edu/embedded?id=a8161777-6eb8-4087-bb0f-a96ed81f2aaf&siteId=61e0606e-415d-4001-8206-ffde48430c64)

4. Explore and determine how the following Extended Finite State Machine, UserName, works. (Standard FSMs do not include semantic actions used in these examples. The semantic actions are labeled in the diagram as “[n]”, and there is a corresponding specification of what actions should occur should that labeled transition occur.)



Basic UserName Recognizer

Semantic Actions:

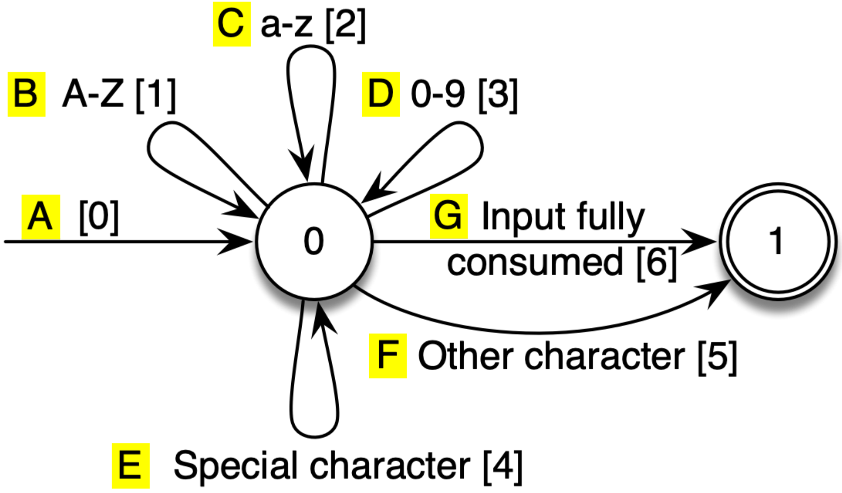
```
[0]   1. Set SizeCounter to zero

[1]   1. Increment SizeCounter

[2]   1. If SizeCounter < 4
             issue too short error message
         else if SizeCounter > 16
             issue too long error message
```

UNChar is one of these:
A - Z, a - z, 0 - 9

The special character period may only be used between two UNChar characters

5. Produce a new FSM diagram that requires the first character of the UserName to be alphabetic (as opposed to alphanumeric). In addition, allow minus signs and underscore as well as period characters **between** subsequent alphanumeric characters. You are encouraged to use Astah to produce this diagram, as it will be required in the future, but it is not required for this assignment. Be sure to define any new terms that you include in your diagram. You are not required to provide the yellow highlighted transition labels in your new diagram or replicate the semantic action definitions (e.g., [1]).
6. Duplicate the Foundational Code application and name it HW1.
7. Modify the UserNameRecognizer class inside of UserNameRecognizerConsoleTestbed so that it uses the new FSM diagram and issues helpful error messages when the input is not recognized. The new code must be in the same style as the original code. Be sure to document your new code in a manner consistent with the internal comments from the provided code.
8. Study the following FSM and update HW1 so it implements the following FSM recognizer and issues helpful error messages when the input is not recognized by the FSM. The new code must be written using the same style as the original code. Be sure to document your new code in a manner consistent with the internal comments from the provided code.

Basic Password Evaluator



A special character is one of these:

~ ` ! @ # $ % ^ & * ( ) _ - +
{ } [ ] | : , . ? /

Boolean Flags:

1. upperCase
2. lowerCase
3. numericChar
4. specialChar
5. longEnough
6. otherChar

Semantic Actions:

[0] 1. Set charCounter to zero
2. Set all Boolean flags to False
3. Set currentChar to first input char

[1] 1. Increment charCounter
2. Set upperCase to True
3. Set currentChar to next input char

[2] 1. Increment charCounter
2. Set lowerCase to True
3. Set currentChar to next input char

[3] 1. Increment charCounter
2. Set numericChar to True
3. Set currentChar to next input char

[4] 1. Increment charCounter
2. Set specialChar to True
3. Set currentChar to next input char

[5] 1. Set otherChar to True

[6] 1. If charCounter >= 8
Set longEnough to True

1. Test the HW1 you have produced to make sure that it satisfies the new requirements and does not accept UserNames or Passwords that would be rejected by the two FSMs.
2. Produce a PDF that contains the new FSM diagram produced in step 5, a set of screenshots showing the source code and comments for your new UserNameRecognizer class, the code for the new PasswordRecognizer class, screenshots showing the results of each of the test cases, and explanatory text for each that explains what was being tested and why the output supports your assertion that your code is working properly.
3. 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**!

Deliverables

A PDF document must be produced that covers the following items.

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

  HW1 Assignment Template.docx (https://canvas.asu.edu/courses/215181/files/100818760?wrap=1)

- Cover page complete with your name. (5%)
- Task 5: Extended Finite State Machine diagram produced as required. (10%)
  - allow minus signs and underscore as well as period characters **between** subsequent alphanumeric characters (5%)
  - other than above, the original functions of the FSM are properly represented (3%)
  - no extraneous items have been added (2%)
- Implementation of the Task 5 Extended Finite State Machine alights with the diagram and is documented, making clear what code supports which element of the FSM. (25%)
  - The new requirements have been properly implemented. (10%)
  - The documentation of the new code properly describes the new code. (5%)
  - The code and documentation align with the rest of the code and documentation. (5%)
  - No extraneous items have been added or errors introduced. (5%)
- Implementation of the Task 8 Extended Finite State Machine alights with the diagram and is documented, making clear what code supports which element of the FSM. (25%)
  - The new requirements have been properly implemented. (10%)
  - The documentation of the new code properly describes the new code. (5%)
  - The code and documentation align with the rest of the code and documentation. (5%)
  - No extraneous items have been added or errors introduced. (5%)
- Testing of the two implementations has been performed, the results and the explanatory text show the code is working, and the resulting screenshots are as described in Task 10. (20%)
  - The testing covers all the transitions and states of the two FSMs. (10%)
  - The screenshots and the text make it easy to see that the code is working. (5%)
  - The testing code and documentation align with the rest of the code and documentation. (5%)
- The grader must be able to examine the code in the PDF that you submitted, 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%)

Rubric

Criteria	Ratings			Pts
Template Usage: The pdf must follow the provided template format	**5 pts** **Correct template used and fully filled with accurate content.**	**2.5 pts** **Template used but contains minor inaccuracies or missing elements.**	**0 pts** **Incorrect or missing template.**	5 pts
Cover Page	**5 pts** **Name and required details are present**	**2.5 pts** **Minor errors in the cover page.**	**0 pts** **Cover page missing.**	5 pts
The new FSM Diagram satisfies the new requirements	**10 pts** **Fully satisfies the requirements**	**5 pts** **Minor errors in the FSM diagram**	**0 pts** **FSM diagram missing or incorrect.**	10 pts
Task 5 Implementation: The new requirements have been properly implemented	**10 pts** **All the requirements have been fully implemented**	**5 pts** **Partially implemented or minor errors with the implementation.**	**0 pts** **Not implemented or not working.**	10 pts
Task 5 Documentation: The documentation of the new code properly describes the new code.	**5 pts** **Documentation properly describes the code.**	**2.5 pts** **Lacks some details in the documentation.**	**0 pts** **Documentation is missing or unclear.**	5 pts
Task 5 FSM Diagram Alignment: The code and documentation align with the rest of the code and documentation	**5 pts** **Seamless integration with original code and documentation.tation properly describes the code.**	**2.5 pts** **Minor inconsistencies with the original code.**	**0 pts** **Documentation does not match code.**	5 pts
Task 5: No extraneous items have been added or errors introduced.	**5 pts** **No unnecessary additions or errors.**	**2.5 pts** **Some extraneous elements present.**	**0 pts** **Significant extraneous elements added.**	5 pts
Task 8: The new requirements have been properly implemented.	**10 pts** **All requirements are implemented.**	**5 pts** **Some requirements not satisfied or any other minor issues**	**0 pts** **None of the requirements are satisfied**	10 pts

Criteria	Ratings			Pts
Task 8: The documentation of the new code properly describes the new code.	**5 pts** **Important parts of the added code are documented**	**2.5 pts** **Some documentation is missing**	**0 pts** **Significant parts of newly added code are missing documentation**	5 pts
Task 8: The code and documentation align with the rest of the code and documentation.	**5 pts** **Added documentation and code aligns with the existing style**	**2.5 pts** **Most of the added documentation and code aligns with the existing style**	**0 pts** **Added documentation and code has a major deviation from the existing style**	5 pts
Task 8: No extraneous items have been added or errors introduced.	**5 pts** **No unnecessary additions**	**2.5 pts** **Some unnecessary additions or issues**	**0 pts** **Significant unnecessary additions or issues**	5 pts
The testing covers all the transitions and states of the two FSMs.	**10 pts** **Covers all transactions and states of Both FSM'S.**	**5 pts** **Some of the transactions or states are not covered in testing.**	**0 pts** **No Testing Done.**	10 pts
Testing: The screenshots and the text make it easy to see that the code is working.	**5 pts** **Screenshot clearly covered all the testing**	**2.5 pts** **Screenshot does not cover everything that is required.**	**0 pts** **No Screenshots**	5 pts
Testing: The testing code and documentation align with the rest of the code and documentation.	**5 pts** **Testing Code Align with the Documentation.**	**2.5 pts** **Documentation lacks clarity.**	**0 pts** **No Documentation.**	5 pts
Updated code in the pdf formatted with internal documentation, and determine that it is consistent with the provided code	**10 pts** **Well-formatted, internally documented, and consistent with the provided code.**	**5 pts** **Formatting and documentation present but inconsistent.**	**0 pts** **Missing Code**	10 pts
Bonus points	**0 pts** **Full Marks**	**0 pts** **No Marks**		0 pts

Criteria	Ratings	Pts
		Total Points: 100