Mohammed Uddin

CS6301

SDLC Documentation for LinkedList to SortedStack Project

Below is the linked list to stack SDLC process. Please note I used the format my employer uses (Based on Agile).

**Planning**

1. *Project Initiation*:

Objective:

Modify an existing Java program that reads a list of integers from the user, stores them in a stack, and prints the sorted stack as opposed to a LinkedList

Key Features / **Requirements**:

* Utilizes a Stack data structure.
* Sorts using the Collections.sort method.

2. *Sprint Planning:* ***Design and Prototype***

Sprint Goals:

* Implement input reading functionality.
* Develop sorting functionality.
* Implement output printing.

User Stories:

* As a user, I want to input a list of integers.
* As a user, I want the integers to be stored in a stack.
* As a user, I want the stack to be sorted in ascending order.
* As a user, I want to see the sorted stack.

3. *Implementation:*

Tasks:

* Create the readInput method to read integers from the user and store them in a stack.
* Implement the sortStack method to sort the stack using Collections.sort.
* Develop the printSortedNumbers method to print the sorted stack.
* Create the main method to execute the program.

4. *Sprint Review:*

Review:

* Check and Verify that the program reads input correctly.
* Ensure the stack is sorted as expected.
* Confirm the program prints the sorted stack correctly.

5. *Sprint Retrospective:*

Positive Outcomes:

* Successful implementation of input reading.
* Sorting functionality worked as expected.
* Accurate printing of the sorted stack.

Areas for Improvement:

* Potentially add handling and support of non-integer inputs, especially floats.

6. *Iteration:*

Feedback Incorporation:

* Address any issues or feedback from the review.
* Consider implementing enhancements or optimizations.

7. *Release /* ***Testing*** */* ***Deployment****:*

Release Criteria:

* All user stories implemented.
* Code reviewed and tested manually.
* Documentation updated.

8. *Documentation:*

Code Documentation:

* Ensure code comments are clear and explain key functionalities.
* User Documentation:
* Create a README file explaining how to use the program.

9. *Continuous Improvement /* ***Operational Maintenance*** *:*

Feedback Loop:

* Gather feedback from users and team members.
* Consider feedback for future iterations.