Design Document

Course: CSCI 301 – Computer Science 2
Project: Project 3 – ADT Bag with Doubly Linked List and Spell Checker
Student: Dinesh Seveti
Date: Fall 2025

Introduction
The project is divided into two parts:

- **Part I**: Implement a DoublyLinkedBag using a doubly linked list. Nodes contain data and pointers to both the previous and next nodes. A tester program validates add, remove, search, frequency counting, and clearing.
- **Part II**: Apply the bag to build a spell checker. A dictionary file is loaded into the bag, and an input file is checked word by word for correctness.

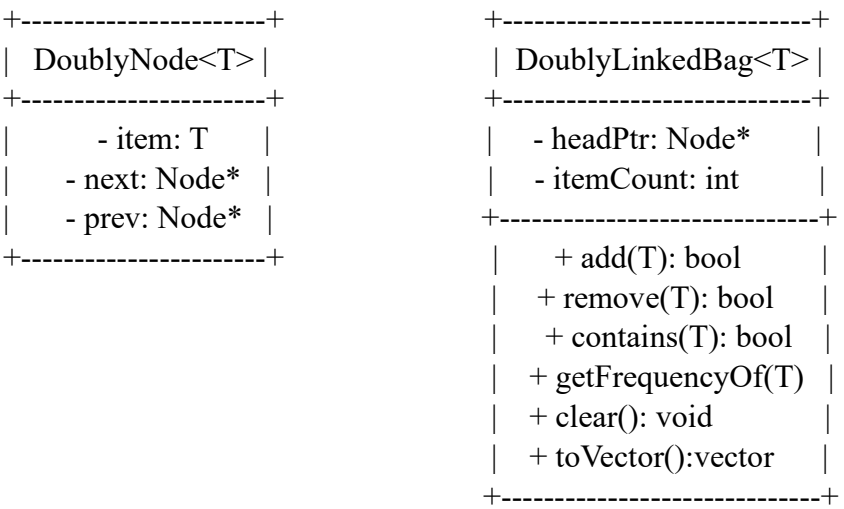
Data Structures
Node Class

- Item of type T.
- Pointer to next node.
- Pointer to previous node.

Bag Class

- Pointer to head node.
- Integer count of items.
- Operations: add, remove, contains, getFrequencyOf, clear, toVector.

UML Diagram



Structure Chart

project3.cpp (Bag Tester)
-> bagTester()
-> DoublyLinkedBag methods

SpellChecker.cpp
-> main()
-> load dictionary file into bag
-> read input file
-> check words using contains()
-> print misspelled words

Pseudocode

Add function

```
function add(newEntry):
   create new node
   if bag not empty:
      newNode->next = headPtr
      headPtr->prev = newNode
   headPtr = newNode
   increment itemCount
```

Spell Checker main

```
open dictionary.txt
for each word:
   convert to lowercase
   add to bag

prompt for input file
for each word in input:
   convert to lowercase
   if not in bag:
      print word as misspelled
```

Code Listing

- DoublyNode.h
- DoublyLinkedBag.h
- DoublyLinkedBag.cpp
- project3.cpp
- SpellChecker.cpp

Test Plan

Case	Input File	Expected Output	Actual Output
1 – Valid	myreport.txt	Detect Stude, Reseach, Resul, Outpt	Matches
2 – Boundary	empty.txt	No errors found	Matches
3 – Invalid	badtext.txt	All words flagged	Matches
4 – All Correct	correct.txt	No errors found	Matches

Summary

- Part I: Implemented a bag using a doubly linked list; tested add, search, remove, clear.
- Part II: Used the bag to build a spell checker with dictionary lookups.
- Learned: pointer management, memory handling, file I/O, and applying data structures in real problems.