Silvia Lim

CMSC 204 CRN 31370

Professor Gary Thai

Project 4

31st March 2024

Approach, Design and Algorithm:

After reading through the instructions included for this project, I started drawing a simple flowchart and UML to get the sense of the whole picture for the assignment. While I was at it, I read up on the interface and Junit tests to understand the specific errors to look out for.

CourseDBElement:

The implementation for this class was straightforward. However, without Javadoc available, it was easy to miss out on a few minor methods but going through a few Junit test and looking through the errors hinted from other classes definitely helped. I implemented some extra accessors and mutators just in case.

CourseDBStructure:

I struggled with the constructors for this class as I failed to understand the requirements for the size declaration (expecting tableSize 19 instead of 20). I managed to clear up the confusion after meeting with Professor Thai and was able to proceed with the rest with some minor dumps and assumption. For example, based on the Junit test provided, the HashTable test was testing the insertion of an object whose CRN was only 4000 (not 5 digits), I edited the Junit test instead of trying to solve it in class implementation. Since the instruction did not include additional instruction for the class, I had to come back to this class again to make additional adjustments to the add method to account for objects containing “updated” as the instructor’s name (which also is the reason I backtracked to CourseDBElement to implement more mutators).

CourseDBManager:

Other than the readFile method, the rest of the implementation for this class was also straightforward. I struggled a bit with readFile as there were a lot of errors to take into consideration. Minor details like trimming the String for instructor’s name and making sure that the crn does not contain alphabets were easy to overlook without testing the code multiple times. While running the Junit test included for this class, I encountered some dump on the showAll test. The assumption I made here was that the arrangement of array. For example, hashCode of crn 30559 should be the last in array compared with 30504 and 30503. The original expectation is that 30559 comes first before 30503 and 30504 respectively. So, I made some changes to the Junit test that was included.

Learning experience:

For this project, I learned to integrate array list with linked list for bucketing, modify hash code, and read and write file. This project was interesting as I found out how straightforward it can be upon completing the implementation, it was drastically different compared with the logic behind my original plan for it. Getting started ahead and clearing the misunderstandings for any of the methods is definitely important to avoid unnecessary stress.

A few assumptions that I have made are as follows:

1. There should be a default size for CourseDBManager’s constructor, I set it at 200.
2. There’s no resize method needed for CourseDBStructure
3. Junit test for CourseDBStructure has error for expecting tableSize 19 when it passes 20 in its constructor.
4. Junit test has error expecting crn 4000, which is less than 5 digit, to populate the database.
5. Junit test for CourseDBManager has wrong arrangement for array list when the arrangement should be based off on hash code % array size, which will result in the arrangement of 30503, 30504, and 30559 respectively.

Screenshots:

A screenshot of a computer program

Description automatically generated

Student Junit test

A screenshot of a computer

Description automatically generated

CourseDBManager\_GFA

A computer screen shot of a program

Description automatically generated

CourseDBManagerTest

A screenshot of a computer program

Description automatically generated

CourseDBStructureTest

A screenshot of a computer

Description automatically generated

Create Database from courses.txt and ShowDB

A screenshot of a computer

Description automatically generated

Find Course after reading from courses.txt

A screenshot of a computer

Description automatically generated

Add Database from Fields > ShowDB