**CSCE 4523 Database Management Systems**

**Homework 1**

**Due: Monday, February 3 at 11:59pm**

**By: Scott Jackson**

**Nick Hester**

**Objectives**

The objectives of this homework were to implement a database using a sorted file of fixed length records. The data that we used came from the file Fortune\_500\_HQ.csv. The overall goal of this assignment was to understand and practice using file management techniques to implement a database system. Our file-based database needed to be able to handle overflow by writing records to an overflow file which contained unsorted records. At the point of 4 or more records in an overflow file, our program was supposed to reorganize the database so that the overflow records were merged back in with the sorted records.

**Approach**

We implemented this by looking over the instructions, the broke it down by first reading in the file. Once we read in the file correctly, we then moved on to the binary search and getRecord to make sure we could properly search through the data. The language that we chose to use was C++ and our strategy was to first create the menu, secondly, we added the additional files necessary, thirdly we implement all of the different classes in our DatabaseSelection.cpp, and finally we filled in each of the classes in the DatabaseSelection.cpp. The format we chose for the record was to search via primary key. Our primary key was the company name. The size of the fields was 6. The total record size was 500. We used the config file to save the number of records in the data file. It also describes the names and sizes of the fields in order. The sample code we used was the “Reading and writing fixed length files in C++”. We handled the overflow by only allowing four records into the overflow, and once the overflow was full, our program was supposed to reorganize the data so that the overflow records were merged back in with the already sorted records.

**Results**

The only error handling we had was covered in the menu. Whenever someone would select a number that wasn’t in between the boundaries, it would ask for the number again. The menu worked well and reading in the files worked well. We had some trouble with the binary search locating the correct record, which in turn skewed the rest of our project. We both did an equal amount of coding and problem solving.

**Testing**

We tested our program by searching for specific companies to see if it would add them to the overflow. Upon testing to see if we could locate a certain company, it would output “Record was not found”. Our menu and reading in the files worked, but searching for a company would not output anything other than “Record was not found”. There weren’t any additional test runs because we were unable to get the binary search working correctly.

**Typescript**

**Additional Comments**

I (Nick Hester) felt as if this project was a little too much for the first homework assignment. I don’t know if I’m just behind, but I feel like I haven’t learned how to do most of this. I feel as if this project should have been the end goal rather than the beginning because this made me completely inadequate. I feel as if we could have built up to this, incorporating different parts of this project and eventually reaching a point to include all of the parts into one project. This project just made me feel completely lost and added an incredibly amount of stress, especially for it being the first homework assignment.