CSU22012 Final Project

Design Document

Paddi Hua

20333316

**Introduction:** This is the design document for my bus management system in course CSU22012, Algorithms and Data Structure. The main goal of this program is to use the data provide and help the user to get different information and suggestions from the data, e.g. which way to get from a stop to another stop is shorter.

**Section 1:**

For this part I first load all the useful information in three txt file in to the hash maps with key of I (number sequences), using three while loop and three scanner. Load the needed information into the two array to use in the shortest path. When loading the cost in transfers file I add an if statement to check is the length of the string equal to 3, if it equal to 3 this mean we don’t have a minimal transfer time (cost = 2) so we skip this section. If is not equal to 3 then we add the minimal transfer time to the hash map.

**Section 2:**

For this part the main thing is how to use TST to print out all the stop information needed. I create a new class call busStop to implement the whole thing using TST. I got the whole TST class from google (reference at the end). First I use a string to read in each line of the stop file and split using .split function. This allows me to move FLAGSTOP, NB, SB, WB, EB to the end of the string and I use a string builder to form my new string and put the details we need to print inside the new string. Then I put this string into a map and set the stop ID as the key so when I need this information I can just take everything out using the key, then print it out using my print function which also use TST.

**Section 3:**

This part is similar to section 2 but I didn’t use TST, instead I write a new class to print out each line of the string as it is in the file. First create two array list, one contains all the information from the file and second one contain the information we need to output. Then we check is the user input matches with numbers 0-9 to check is it a valid input, if it is we compare this value to each line of our input file to see which line matches the arrival time we want. Before this you also have to split the string in two ways, because if user input “5:25:00” it is not going to match with the time we have because it actually “05:25:00” with a space at the front. I use a method called SimpleDateFormat to compare the user input time with 10:00:00, if is smaller then it means the hours is a single number, e.g. 5:25:00, then we need to split the string using “ ,” instead of “,” to get away the space, if is bigger than 10:00:00 then it means the hour has two number, so we just split it as normal with “,”. After this is pretty easy, just add the time we got that matches the time user input into the array and use a print function to print it out. In the print function I also add the title just because is easier to see in user interface.

**Section 4:**

This is the front end part of the program. For this I just use the terminal to input and output all the information. I decide to use a while loop so this can keep running until the user quit the program by themselves. This can be easily done just by set a Boolean value. They I give user four options, 0 to quit the program, 1 for section 1 (shortest path), 2 for section 2 (search bus stop by name), 3 for section 3 (search by arrival time). When user input 0 then set the Boolean value to true to end the while loop and the program. 1 we take the two input and enter the Dijkstra part, 2 we take the input using a scanner and we turn the input into all upper case because is all upper case in the file. Then we just call the class we write and print out all the stops matches. 3 take the user input using .trim and call the class to search arrival time. I also use try and catch to catch any exceptions occur, and all the print we need to communicate with the user.

**References:**

https://algs4.cs.princeton.edu/52trie/TST.java.html

<https://github.com/Rodeby/CSU22012>

**GitHub link:**

https://github.com/hua1013/CSU22012-final-project