

## CMPE1101 Term Project Report

### 1. General Purpose of the Program

“The Bus Trip”, is a Command Line Interface (CLI) tool developed for small business bus firms. It digitalizing and making easy and fast to trip administration and ticket sell. Main purpose’s is digitalize manuel processes such as creating trip, selling and cancelling ticket. Target users are terminal users who don’t need to use complex interfaces.

### 2. Program Flow

Program serves a menu with infinite while loop to user. Depends on the choice the menu redirect to the related function. When a trip created, it saved as “trips.tbt” file. Updating and deleting trip functions performed with creating a “tmp.tbt” temporary file with filtration and then swapping names between trips.tbt and tmp.tbt.

### 3. Data Structures

The fundamental data structure is “struct Trip” block. All the properties of the trip including Trip ID, departure point, arrival point, trip date, trip time, driver’s full name, licence plate, number of seat and number of sold seat are stored in “Trip” struct.

### 4. Functions

create\_trip, creates a struct and trips.tbt file with given struct input.

list\_trip, lists all the trips with formatted string from trips.tbt.

query\_trip, queries a trip with given trip ID input.

update\_trip, updates struct and trips.tbt file with given struct input.

delete\_trip, deletes a trip with given trip ID input.

sell\_ticket, sells a ticket, updates number of sold seat in trips.tbt and creates a receipt file.

cancel\_ticket, cancels tickets and updates number of sold seat in trips.tbt

create\_receipt, creates a receipts with given trip ID, passenger name and citizen ID input.

check\_ID, checks if given trip ID exists in trips.tbt.

print\_banner, prints menu and redirects to related function with given number input.

### 5. File Operations

Data’s processed in binary mode with wb, ab, rb. Struct’s are directly writed to the trips.tbt and tmp.tbt files. Receipt is in readable text format.

### 6. Error Handling

File opening handling performed with NULL check. Handled ID collision with check\_ID. Detailed error handling messages throwing with [ERR] tag. The user informed with [INF] tag in some operations.

### 7. Dependencies

Written with only standart libraries. stdio.h, for input/output operations, stdlib.h for running OS operations and string.h for string processing.

### 8. Security

Prevented buffer overflow with scanf limitation.

### 9. Conclusion

“The Bus Ticket”, is a functional prototype that performs basic Create, Read, Update, Delete operations. Proper for basic bus and ticket register needs.