# CAPSTONE PROJECT: BUS RESERVATION SYSTEM

## AHJIN RESERVATION

### 1. Introduction

Ahjin Reservation is a simple C program that implements the aspects of a modern bus booking system.

### 2. Overview

This is a data flow diagram (DFD) showing the overview of the program;

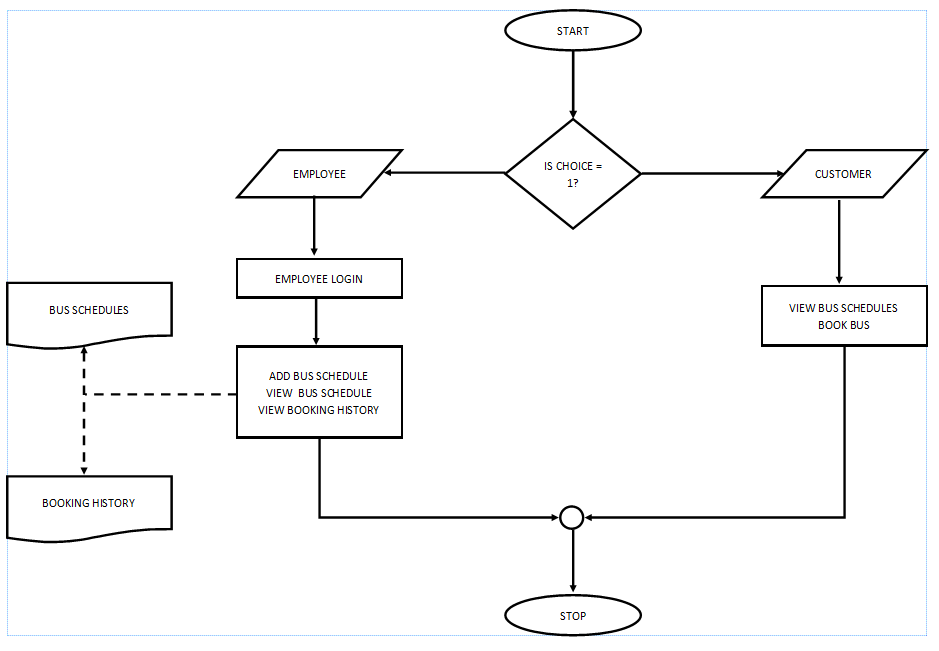


Figure 2.1

### 3. Usage

Once you run the program, the first thing is to know which group of users you belong to, i.e., **customer** or **employee**. This is due to the fact that the program will run work differently for both groups of users.

For the customer group, should you want to book a bus but you don’t know which buses are available, pick the option **View all bus schedules**. This will display a list of the available buses with their details such as price, source and destination, time, etc. Once you are sure of the bus you want, press any key to go back to the menu then pick the **Book a seat** option. You will be prompted for necessary input such as name and phone number. After the booking is successful, a receipt will be displayed which will serve as proof of a successful booking.

For the employee group, before you are able to do anything, you are required to login first by inputting the username and its corresponding password. A maximum of 3 trials are given otherwise the program closes. For security purposes, passwords are masked with an asterisk (\*) during input. Once the login is successful, the employee has access to the program and is able to:

* Add new bus schedules
* View existing bus schedules
* View booking history

In the case one needs to modify or completely erase a bus schedule, they can do so in the **Bus Schedules.txt** file is where the records of the bus schedules are stored.  
Similarly, if an employee calls the customer care services wanting to cancel a booking, an employee can erase the customer from the **Booking History.txt** file and update the seat in the **Bus Schedules.txt** file.

### 4. Features

1. **Bus Schedule Management:**

* Add new bus schedules with details like source, destination, time, and fare.
* View a comprehensive list of all available bus schedules.

1. **Customer Interaction:**

* Enable customers to book seats on buses.
* Display available seats and allow customers to choose from them.
* Generate and show a booking receipt with essential details.

1. **Employee Access:**

* Implement secure employee login functionality.
* Allow employees to add new bus schedules.
* Provide employees with the ability to view booking history.

1. **Persistence:**

* Save and load bus schedules to/from a file for data persistence.
* Maintain booking history for reference and record-keeping.

1. **User-Friendly Interface:**

* Implement a clear and intuitive menu system for easy navigation.
* Display relevant prompts and messages for user guidance.
* Ensure a smooth and hassle-free experience for both customers and employees.

### 5. Code Structure

1. **Structures:**

* Use structures (Employee, Bus, Customer) to organize related data.
* Each structure encapsulates specific information relevant to its purpose.

1. **Functions:**

* ***main:*** Orchestrates the program flow and user interactions.
* ***employeeLogin:*** Validates employee credentials for secure access.
* ***addBusSchedule:*** Gathers information to add a new bus schedule.
* ***viewBusSchedules:*** Displays a list of all available bus schedules.
* ***saveBusSchedulesToFile*** and ***loadBusSchedulesFromFile:*** Handle data persistence as well as save and load data from files.
* ***bookSeats:*** Manages the booking process, seat selection, and receipt generation.
* ***getCurrentTime:*** Retrieves the current time for timestamps.

### 6. Conclusion

This concludes the documentation of the Ahjin Reservation program.  
Thank you for reading this program documentation.