**TRAVEL INC.**

Travelling is something everyone is fond of. This project aims at developing a platform for convenient travel planning and bookings. It focuses on collecting and managing the data of various flights, trains and hotels, this is done by the implementation of various data structures. The program will house all details about train and flight schedules, and the availability of rooms in various hotels and will enable the user to book tickets for their preferred mode of travel and even make reservations in their chosen place of stay subject to availability. It offers its customers a digital wallet to store money and make quick payments. It requires users to create password protected accounts; that store information such as their name, mobile number, E-mail address, wallet balance, bank account details and net banking details; in order to carry out day to day transactions in a safe and convenient environment. The program stores a database of all its customers which contains all information stored by each account, with the application of file handling. The customer can edit most of this information, barring some invariable details, at any point in time. Due to lack of connectivity to the actual bank database at the current stage, the program creates a virtual bank database which requires the user to input his/her bank balance manually. The bank balance of the registered customers can be monitored and updated by a bank employee by logging in with the unique Bank User ID ( currently set as “bank1234” ) and password ( currently set as “1234bank” ). It provides a separate login for administrators that enables registration of new flights, trains and hotels ( current Admin User ID: admin and Password: password@admin). The program also provides a procedure to check the connectivity between two given destinations ( i.e. whether a given destination can be reached from the given starting point ). This enables users to plan and execute their travel and stay in an easy and convenient manner.

# **DATA STRUCTURES USED**

## Graph: To check connectivity of two places

## Map: To implement the graph

## Vector: To store list of destinations connected to source

## Queue: To implement BFS of graph

**MAIN FUNCTIONS**

* All get and set functions in every class for value of private data members
* View\_(flight/train/hotel)\_details-to view details of all the registered entities
* Enter\_(flight/train/hotel)\_details-to enter the details of any entity.
* Train/flight/hotel reservation()-to make reservations
* Cancellation(); to cancel reservation
* Menu(): to display choice
* Mapping():Creates graph to connect source to all connected destinations
* Searchs() and bfs(): BFS traversal for graph to check for path between two places
* Updatebankbalance(): to update bank balance
* flight/train/hotel::delete1():to delete a specific entry based on the respective flight/hotel/train no
* Void user::deactivate()-this is to delete the data of user from the file
* void Deposit(void) – it is to add money to wallet from bank account
* Checkemail() and checkmobile()-there are to confirm whether the email and mobile entered by user is correct or not
* tickets::display()-Displays all reservations made from the user’s account

Name of classes:

* Admin
* Pb:passbook
* User
* Flight
* Travel
* Train
* Hotel
* Tickets
* Bank