```c

/* Project created by Sara Marfella IST188316 on May 16, 2017 */

#ifndef dataManager_h
#define dataManager_h

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>
#include <string.h>

// CONSTANTS
#define MAX_SIZE 150    // max size of a string
#define ID_SIZE 7       // size of Bike ID and station name
#define REGISTERED 1    // the registered member
#define CASUAL 0        // the casual user
#define MALE 1          // the male gender
#define FEMALE 2        // the female gender
#define EXISTING 1      // the existing station status
#define REMOVED 0       // the removed station status

//ALL THE STRUCT

// define a type of date
typedef struct{
    int month;
    int day;
    int year;
    int hour;
    int minute;
}Date;

// Linked List of Trips
typedef struct Trip{
    long int id;
    int duration; // in seconds
    Date start;
    int id_start_station;
    Date end;
    int id_final_station;
    char bike[ID_SIZE];
    int type; // the user can be casual or members
    int year_birthday; // only in case of members
    int gender; // female or male only in case of members
    struct Trip *next;
}Trip;

// Linked List of Stations
typedef struct Station{
    int id;
    char name[ID_SIZE];
    char full_name[MAX_SIZE];
    char municipal[MAX_SIZE];
    double latitude;
    double longitude;
    int status; // existing or removed
    int max_bikesIn;
    int min_bikesIn;
    int max_bikesOut;
    int min_bikesOut;
    float avg_bikesIn;
    float avg_bikesOut;
    struct Station *next;
}Station;

```

```
67
68
69
70  // LinkedList of Routes
71  typedef struct Route{
72      int total;
73      int id_start_station;
74      char name_start_station[ID_SIZE];
75      char id_final_station;
76      char name_final_station[ID_SIZE];
77      struct Route *next;
78  }Route;
79
80  // FUNCTIONS
81
82  // file readers
83  Station * readStationData(char*);
84  Trip * readTripsData(char*);
85
86  // list creators
87  Route * createRoutesList(Trip*, Station*, int);
88  Station * countBikes(Trip*, Station*, int, int);
89
90  // list filters
91  Trip* selectTripsByTime(Trip*, int, int);
92  Trip* selectTripsByDuration(Trip*, int);
93  Trip* selectTripsByDay(Trip*, int);
94  Trip* selectTripsByIdStation(Trip*, int);
95
96  // helpers
97  Trip* copyTripToList(Trip*, Trip*);
98  int calculateWeekDateFromDate(int, int, int);
99  void sortedInsert(Route**, Route*);
100 char * getStationNameById(int, Station*);
101
102 #endif /* dataManager_h */
```