

```

1  /*  GENERAL INFORMATION
2  *
3  *  Project created by Sara Marfella IST188316 on May 16, 2017
4  *
5  *  Implemented:
6  *  1. Load the data structures required to represent
7  *     the stations and the trips.
8  *  2. Read parameters of main and 2 files required
9  *     to make the program work.
10 *  3. Build 2 menus and data input.
11 *  4. Build the lists of trips and stations,
12 *     inserting items in an ordered way.
13 *  5. Implementation of trips list mode.
14 *  6. Implementation of data filters.
15 *  7. Implementation of stations list mode.
16 *  8. Implementation of routes list mode.
17 *
18 *  Not Implemented:
19 *  9. Implementation of graphic mode.
20 *  10. Implementation of statistical analysis.
21 *  11. Improvements of application graphics mode,
22 *      with a real map.
23 */
24
25 #include <stdio.h>
26 #include <stdlib.h>
27 #include <time.h>
28 #include <stdbool.h>
29 #include <string.h>
30 #include "dataManager.h"
31 #include "print.h"
32
33 // EVIL GLOBAL VARIABLES
34 int filter_hour_start = -1;
35 int filter_hour_end = -1;
36
37 char trips_file[MAX_SIZE];
38 char stations_file[MAX_SIZE];
39
40 // DECLARATION OF FUNCTIONS
41
42 // command line interface
43 void mainMenu(Trip*, Station*);
44 void listOfStationsMenu(Trip *, Station *);
45 void selectDataMenu();
46 void selectStationMenu(Station*, Trip*, int);
47 void statsMenu();
48 void clearInputBuffer();
49
50 /* Main Function
51 * \param mode and file (trips file  and stations file)
52 */
53 int main (int argc, char *argv[]){
54
55     // check for correct input
56     if (argc != 4) {
57         printf("Error - required arguments missing.\n");
58         printf("Please start with -t trips-filename stations-filename\n");
59         return 0;
60     }
61     if (strcmp(argv[1], "-t") != 0) {
62         printf("Sorry - only text mode is available,");
63         printf("Please start with -t\n");
64         return 0;
65     }
66

```

```

67     strcpy(trips_file, argv[2]);
68     strcpy(stations_file, argv[3]);
69
70     Trip * allTrips = readTripsData(trips_file);
71     Station * allStations = readStationData(stations_file);
72
73     mainMenu(allTrips, allStations);
74
75     return 0;
76 }
77
78 /* Command Line Interface */
79
80 /* mainMenu: prints the main menu and handles the main menu options
81  * \param tripList      the header of the trip list
82  *                      (can be the filtered list)
83  * \param stationsList  the header of the stations list
84  */
85 void mainMenu(Trip * tripsList, Station * stationsList){
86     int command;
87     printf("\n * M A I N   M E N U * \n\n");
88     printf(" [ 1 ]   Select the data\n\n");
89     printf(" [ 2 ]   Print List of trips\n\n");
90     printf(" [ 3 ]   Print List of stations\n\n");
91     printf(" [ 4 ]   Print List of routes\n\n");
92     printf(" [ 5 ]   Print List of statistics\n\n");
93     printf(" [ 6 ]   Quit\n\n");
94     scanf("%d", &command);
95     clearInputBuffer();
96     switch (command) {
97         case 1:
98             selectDataMenu(tripsList, stationsList);
99             break;
100        case 2:
101            command = -1;
102            while ((command < 0) || (command > 32000)) {
103                printf("How many trips do you want to print?");
104                printf("0..32000, enter 0 for all)\n");
105                scanf("%d", &command);
106                clearInputBuffer();
107            }
108            printTripsList(tripsList, command);
109            mainMenu(tripsList, stationsList);
110            break;
111        case 3:
112            listOfStationsMenu(tripsList, stationsList);
113            break;
114        case 4:
115            selectStationMenu(stationsList, tripsList, 0);
116            break;
117        case 5:
118            statsMenu();
119            mainMenu(tripsList, stationsList);
120            break;
121        case 6:
122            exit(EXIT_SUCCESS);
123            break;
124        default:
125            printf("Error: invalid command...\n");
126            mainMenu(tripsList, stationsList);
127            break;
128    }
129 }
130
131
132

```

```

133  /* List of Stations Menu:    creates and prints the list of
134  *                            stations with max/min/avg
135  * \param tripList           the header of the trip list
136  *                            (usually the filtered list)
137  * \param stationsList       the header of the stations list
138  */
139  void listOfStationsMenu(Trip * tripsList, Station * stationsList){
140      Station * filteredStations = countBikes(tripsList,
141          stationsList, filter_hour_start, filter_hour_end);
142
143      // option to ask the user: limit (0 for all),
144      //should print stations with no trips (YES, NO)
145      printStationsList(filteredStations, 0, YES);
146      mainMenu(tripsList, stationsList);
147  }
148
149  /* SelectDataMenu: prints and manages the menu to set
150  *                  search criteria
151  * \param filteredTrips       the header of the trip list
152  *                            (can be the filtered list)
153  * \param allStations         the header of all stations list
154  */
155  void selectDataMenu(Trip * filteredTrips, Station * allStations){
156
157      int duration = -1, command;
158
159      printf("\n * Select the mode of your search * \n\n");
160      printf(" [ 1 ]   Period of time (hour start, hour end)\n\n");
161      printf(" [ 2 ]   Day of week\n\n");
162      printf(" [ 3 ]   Max duration of trip (in seconds)\n\n");
163      printf(" [ 4 ]   New Search (reset list)\n\n");
164      printf(" [ 5 ]   Return to Main Menu\n\n");
165      scanf("%d", &command);
166      clearInputBuffer();
167      switch (command) {
168          case 1:
169              command = -1;
170
171              while ((command < 0) || (command > 23)) {
172                  printf("Insert start time of trip (hour 0..23):\n");
173                  scanf("%d", &command);
174                  clearInputBuffer();
175              }
176              // temporary place to store start time cannot
177              //overwrite the global now
178              // because we use it to understand if user has
179              //used the filter before, later in the code
180              int start = command;
181
182              command = -1;
183              while ((command < 0) || (command > 23)) {
184                  printf("Insert end time of trip (hour 0..23):\n");
185                  scanf("%d", &command);
186                  clearInputBuffer();
187              }
188              filter_hour_end = command;
189
190              // if user had filtered the list before,
191              // reset the list to avoid problems
192              if (filter_hour_start != -1) {
193                  // now we can save the start time
194                  filter_hour_start = start;
195                  Trip * allTrips = readTripsData(trips_file);
196                  printf("Note: a time filter was already applied.\n");
197                  printf("The trip list was reset to use the new param.\n");
198                  filteredTrips = selectTripsByTime(allTrips,

```

```

199         filter_hour_start, filter_hour_end);
200     } else {
201         filter_hour_start = start;
202         filteredTrips = selectTripsByTime(filteredTrips,
203             filter_hour_start, filter_hour_end);
204     }
205     mainMenu(filteredTrips, allStations);
206     break;
207 case 2:
208     command = 0;
209     while ((command < 1) || (command > 7)) {
210         printf("\nPlease insert the day of trip:\n\n");
211         printf(" [ 1 ] Monday\n");
212         printf(" [ 2 ] Tuesday\n");
213         printf(" [ 3 ] Wednesday\n");
214         printf(" [ 4 ] Thursday\n");
215         printf(" [ 5 ] Friday\n");
216         printf(" [ 6 ] Saturday\n");
217         printf(" [ 7 ] Sunday\n");
218         scanf("%d", &command);
219         clearInputBuffer();
220     }
221     filteredTrips = selectTripsByDay(filteredTrips, command);
222     mainMenu(filteredTrips, allStations);
223     break;
224 case 3:
225     while ((duration < 0) || (duration > 32767)) {
226         printf("Insert the max duration of trip (in seconds):\n");
227         scanf("%d", &duration);
228         clearInputBuffer();
229     }
230     filteredTrips = selectTripsByDuration(filteredTrips, duration);
231     mainMenu(filteredTrips, allStations);
232     break;
233 case 4:
234     filter_hour_end = -1;
235     filter_hour_start = -1;
236     Trip * allTrips = readTripsData(trips_file);
237     mainMenu(allTrips, allStations);
238     break;
239 case 5:
240     if (filteredTrips != NULL)
241         mainMenu(filteredTrips, allStations);
242     else {
243         Trip * allTrips = readTripsData(trips_file);
244         mainMenu(allTrips, allStations);
245     }
246     break;
247 default:
248     printf("Error: invalid command...\n");
249     if (filteredTrips != NULL)
250         selectDataMenu(filteredTrips, allStations);
251     else {
252         Trip * allTrips = readTripsData(trips_file);
253         selectDataMenu(allTrips, allStations);
254     }
255     break;
256 }
257 }
258
259 /* selectStationMenu:
260  * \param allStations the header of all stations list
261  * \param filteredTrips the header of the trips list (can be filtered)
262  * \param id the ID of the station
263  */
264 void selectStationMenu(Station * allStations, Trip * filteredTrips, int id){

```

```

265     int command = -1;
266     char stationName[ID_SIZE] = "";
267
268     // validate id of station
269     while (strcmp(stationName, "") == 0) {
270         printf("Insert the id of the station:\n");
271         scanf("%d", &id);
272         clearInputBuffer();
273         strcpy(stationName, getStationNameById(id, allStations));
274     }
275     filteredTrips = selectTripsByIdStation(filteredTrips, id);
276
277     command = -1;
278     while ((command < 0) || (command > 32767)) {
279         printf("How many routes do yo want to print? (0 for all)\n");
280         scanf("%d", &command);
281         clearInputBuffer();
282     }
283     printRoutesList(filteredTrips, allStations, id, command);
284     Trip * allTrips = readTripsData(trips_file);
285     mainMenu(allTrips, allStations);
286 }
287
288 /* statsMenu: placeholder for the stats section
289 */
290 void statsMenu() {
291     printf("Sorry, the statistics are not yet implemented\n");
292 }
293
294 /* clearInputBuffer: handles input of multiple characters
295 * resets the input buffer to avoid double-commands
296 * Source: stackoverflow.com/questions/3969871/
297 */
298 void clearInputBuffer() // works only if the input buffer is not empty
299 {
300     char c;
301     do {
302         c = getchar();
303     } while (c != '\n' && c != EOF);
304     return;
305 }

```