C Project : Team 7 - Final report

**Problem Statement:** Develop a datafile of billing information

**Data Structures:**

#define MAX\_CUST 50

#define STRING\_LEN 30

#define MAX\_MONTH 12

#define MIN\_YEAR 1600

#define MAX\_YEAR 2999

#define MIN\_BILLS\_PER\_DAY 20

#define MAX\_BILLS\_PER\_DAY 40

#define MIN\_BILL\_AMT 100

#define MAX\_BILL\_AMT 5000

#define MAX\_INTERVALS 11

#define MIN\_TRANSACTION 1

#define MAX\_TRANSACTION 5

struct date{

    int dd;

    int mm;

    int yy;

};

struct person{

    char name[STRING\_LEN];

double total\_per\_person; // initialized to 0

};

struct item{

char item\_name[STRING\_LEN];

double price;

int initial\_quantity;

int quantity\_sold; // initialized to 0

};

struct details{

int bill\_no;

struct person p;

    double amt; // total amount of a bill

int hr, min; to store time of sale; stored in 24hr format

int no\_of\_items; // number of items purchased by a person; MAX=5

struct item item[MAX\_TRANSACTION]; // max 5 items are bought by a person

};

struct bill{

    struct details d[MAX\_BILLS\_PER\_DAY]; // max 40 bills per day

    int limit; // bill limit for a day; MAX=40

    struct date dt;

double total\_per\_day;

int day;

};

**Functions:**

1. date.h

* int is\_valid\_dates(const struct date\* cur\_date);

// Validate the given date

* int is\_valid\_day(int day);

// Validate the day; 1<day<7; from MONDAY to SUNDAY

* int is\_valid\_range(const struct date\* start\_date, const struct date\* end\_date);

// Compare if end date greater than start date

* int date\_diff(const struct date\* start\_date,const  struct date\* end\_date); //Function to find number of days between two dates.
* void next\_date(struct date\* bill\_date);

// Function to get next date

1. customer.h

* void read\_cust\_data(FILE\* fp, struct person\* p);

// Function to read customer data and store in array of structures

* void disp\_data\_by\_name(const struct person\* p);

// Display data of all persons

1. item.h

* void read\_item\_data(FILE\* fp, struct item\* item);

// Function to read item data and store in array of structures;

* void disp\_data\_all\_items(const struct item\* item);

// Display data of all items sales;

1. bill\_generate.h

* void generate\_bill(struct bill\* b, const struct date\* start\_date, int bill\_day, int days, struct person\* p, struct item\* item);

// Function to generate bill using customer and item data

* static int get\_random\_val(int min, int max, int day)

// Function to generate random number of bills within the range(min, max)

for a given day.

Maximise value for Saturday and Sunday

* static void generate\_array\_of\_unique\_indices(int\* arr, int max);

// Function to get items for a bill from the item set; // 1<=MAX<=5

* static void generate\_bill\_time(int\* time);

// Function to generate sales bill time by randomly incrementing minutes;

time is an array containing hours in time[0] and minutes in time[1]

* static void set\_period(int\* hr1, char\* p1, int\* hr2, char\* p2);

//time is converted from 24hr format to 12hr format;

where p1, p2 is “am” or “pm”

* void disp\_data\_all\_dates(const struct bill\* b, int days);

// Display data of all dates with every bill record and total sales on that day

* void disp\_data\_by\_date(const struct bill\* b, int days, struct date cur\_date);

// Display data of specified day

* void disp\_data\_by\_name(const struct bill\* b, int days, char\* name);

// Display data of specified person

* void disp\_data\_all\_days(const struct bill\* b, int days);

// Display data of all days (Monday-Sunday)

* void disp\_data\_all\_hours(const struct bill\* b, int days);

// Display data of all hours (intervals)

* void disp\_data\_by\_dates\_and\_items(const struct bill\* b, int start, int end, const struct item\* item);

// Display item sales data datewise or periodwise

* void disp\_detailed\_bill(const struct bill\* b, int days);

// Display detailed bill with all information