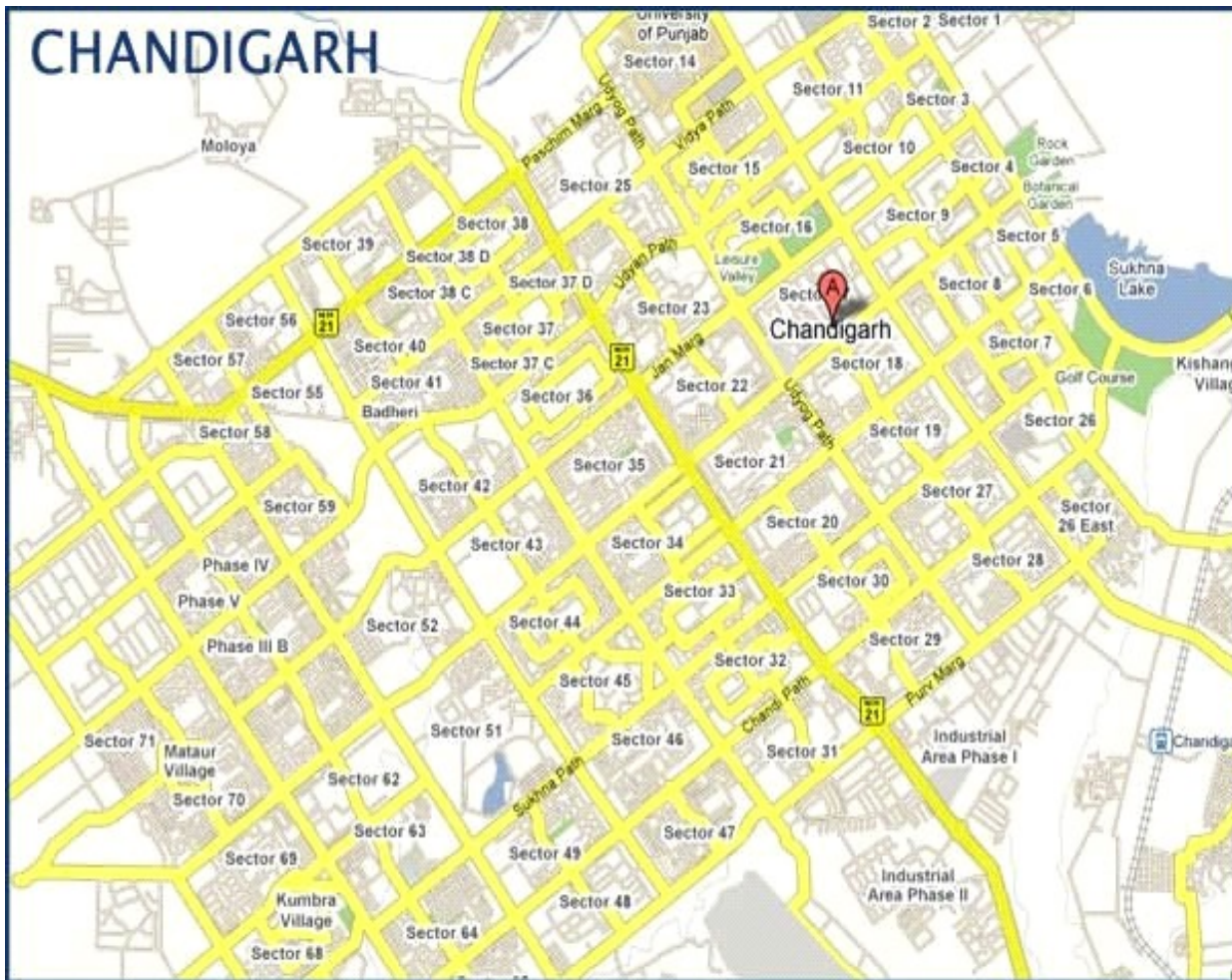


Future Ideas! So where do we use this app?



As visible from the above road map, Chandigarh city seems to be the ideal city to implement this code, and where this code can come use for. The reason: a 2 dimensional linear map!!

- * The data can be extracted from say a GPS system about the position and number of cabs available at any instant, since in the real world the number of cabs at any instant would be different (generated through the concept of random numbers) as there might be cabs either entering or leaving the city from adjoining cities as well. So number of cabs in the city will not actually be a fixed number. In addition to this a time clock could be maintained and it could be pre-assumed at what interval of time there is a red light, or a green light at a given signal at a given checkpoint in a given area and thus the time between the two destinations can be calculated accordingly in a more accurate fashion.
- * In addition to this distances between different stations can be assumed to be different although known initially, and then also a function can be written to calculate a valid shortest route to the destination and fare calculation can be done accordingly.
- * Buses and metros and say ferries (for a city with inland water!!!) for example in Sydney, there is a common travel card that can be used for any mode of transportation and balance is automatically deducted. Hence the consumer can be shown all the different choices he has.
- * On simplification the application would give the following results:
between any two destinations give us the shortest travel route in terms of time. This may also include a break up of the journey for example the best option for a user of the application to go from A to B might be to first take a cab from A to D then take a metro from D to E and finally take a bus from E to B.

Hence, two approaches might be defined:

1. Writing a function that minimizes time
2. Writing another one that minimizes the cost of travel.

- * Another useful thing could be cancellation of an already booked cab.(using threads) For example, a person wishes to cancel the cab he has booked due to some reason. In such a case, a program could be written which cancels his booking , and deducts say a minimal amount of balance for the same.

An application having all the above features would hence make life easy for every kind of consumer ! the one who wants to save his pocket and the one who wants to save time as well!!!

Module Description

of all the functions used in the project.

Functions Used:

- 1)int main()
- 2)int register_acc()
- 3)int login()
- 4)int check_avail(int, int)
- 5)int t_approach(int)
- 6)int fare(int, int, int)
- 7)int time_travel(int, int, int)
- 8)int update(int ,int)
- 9)int bal_update(int)

int main():

The main function contains a switch case which accepts an input from the user. If the input is 1 then it redirects to the login() function, else if it is 2 it redirects to the register_acc() function. If the user makes an invalid login then the login function will run again till the user logs in correctly. After the user registers, the function asks whether the user wants to go back to the main menu or not. This is handled by a do while loop outside the switch case.

int register_acc():

This function is for registering a new user to the database. Details from the user are taken which include their first name, contact number, desired passcode and their desired balance. This program is not responsible for how the user pays their money. Once the user enters and confirms their passcode(which is done by using the strcmp() function of cstring library) then the function generates a unique Customer ID for the user which has to be used by the customer in all their future

transactions. The user has to confirm his passcode within five attempts else the program will exit. After this, all the details of the customer is written into the cust_details.txt file in a comma separated format.

int login():

This function accepts the Customer ID of the user and password. It then searches for the corresponding Customer ID and password in the cust_details.txt file. Once a successful match is found the function displays Welcome <user_name> where <user_name> is the name of the user as present in the file. If a match is not found the function displays invalid login.

int check_avail(int ,int) :

On entering his desired source and destination(passed as the two parameters into the function), the program looks for an available cab in the desired source sector. In the event that a cab is unavailable in the user's desired sector, the program searches for an available cab in the nearest sector. Once the cab has reached the destination(which we have assumed to be instantaneous), the program updates the files cab_data.txt for the changes in cab status. The function returns the distance of the available cab's sector from the user's desired sector.

int t_approach(int) :

Here the parameter is the return value of the function check_avail(). It uses the value of the distance to estimate the time taken by the cab to reach the desired sector of the user. It returns the estimated value of the time to the main.

int fare(int ,int , int) :

The parameters are the desired source sector and the destination sector entered by the user which is passed by the main. The third parameter is the value of flag which indicates whether travel is normal or urgent and calculates the fare accordingly on the given values of S and D. It returns the fare back to the main on computation.

int time_travel(int ,int ,int) :

Here the parameters, similar to the function fare() are S, D and the value of flag. It uses the value of S and D to estimate the time taken to travel the distance for the user. It is this time value(int) which it returns.

int update(int ,int) :

The parameters for this function are S and D. The objective of this function is to update the cab Database in the file "Database.txt". By changing the current sector number of each cab. The program can maintain accountability of each cab's whereabouts and the total cab count.

int bal_update(int) :

The parameter for this function is the fare passed by the main function. It reads the file and reprints the balance with the new balance adjusted for the fare. It uses the global variable c_id which contains the user's cust_id when he logs in and is used to identify the user's account in the file "cust_details.txt".

Serial number	Roll number	name	Design	discussion	documentation	Coding	testing	Miscellaneous
1.	111030006	Akshat Kadam Sanjay	5hrs	10hrs 30mins	-----	12hrs	11hrs	2hrs
2.	111030008	Jaideep Sontakke	2hrs 30mins	8hrs	-----	6hrs	3hrs	2hrs 30mins
3.	111030011	Kshitij Jayakrishnan	3hrs 15mins	7hrs	2hrs	9hrs	6hrs	4hrs 30mins
4.	111030029	Mehak Priya	6hrs	9hrs	11hrs	6hrs 45mins	3hrs	8hrs 30mins
5.	111030026	Mohak Mehta	7hrs 15mins	7hrs 10mins	12hrs	6hrs	2hrs	4hrs
6.	111030014	Mohit Khatri	3hrs	9hrs	1hr	1hr	4hrs 30mins	2hrs
7.	111030002	Md. Tahir Patel	5hrs	9hrs	3hrs	10hrs	8hrs 15mins	9hrs

Status of completion

1. The problem statement which we had decided on earlier stands and we have been able to implement it to our fullest capability considering the time frame we had. However, we were unable to implement a few additional functionalities, namely that of threads which would have made this application practically a lot more feasible by avoiding us the drastic assumption that transactions are instantaneous. It is our unanimous opinion that given a few more days, it would have been implemented.
2. We are hereby enclosing two folders inside the main one, in which one involves execution without implementation of ez windows. However the other one has execution along with ez windows, which due to time constraint as well as a not so useful task of making an onscreen keyboard(which was rejected by the team) runs ,but, with some errors.

PEER REVIEW

NAME	ROLL NO	MARKS
AKSHAT KADAM	111030006	9.5
JAIDEEP SONTAKKE	111030008	8.25
KSHITIJ JAYAKRISHNAN	111030011	9.25
MEHAK PRIYA	111030029	9
MOHAK MEHTA	111030026	9.25
MOHIT KHATRI	111030014	5
MD TAHIR PATEL	111030002	9.25