

Software System for Hailing Cab

UCS2201 – Fundamentals and Practice of Software Development

A PROJECT REPORT

Submitted By

Mugilkrishna D U 3122 22 5001 073

Neha Shanmitha S 3122 22 5001 080

Nikilesh Jayaguptha 3122 22 5001 081



Department of Computer Science and Engineering

Sri Sivasubramaniya Nadar College of Engineering
(An Autonomous Institution, Affiliated to Anna University)
Kalavakkam – 603110

July 2023

Sri Sivasubramaniya Nadar College of Engineering
(An Autonomous Institution, Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this project report titled “**Software System for Hailing Cab**” is the bonafide work of “Mugilkrishna D U (3122 22 5001 073), Neha Shanmitha S (3122 22 5001 080) and Nikilesh Jayaguptha (3122 22 5001 081)” who carried out the project work in the UCS2201 – Fundamentals and Practice of Software Development during the academic year 2022-23.

Internal Examiner

External Examiner

Date:

TABLE OF CONTENTS

S.NO	CONTENT	PAGE NUMBER
1.	Problem Statement	4
2.	Exploration of problem statement	4
3.	Analysis using Data Flow Diagrams	5
4.	Detailed Design	7
5.	Description of each Module	8
6.	Implementation 1. Explanation of how the data is organized and the Rationale behind the selection of a particular language construct 2. Explanation of any other libraries or APIs that have been used 3. User interface design 4. Platform used for Code Development	12
7.	Validation through Detailed Test cases for various scenarios	17
8.	Limitations of the solution provided	26
9.	Observations	26
10.	Learning Outcomes	28
11.	References	29

Problem Statement

Develop a software system for assigning cabs for customers based on their requests and locations.

The customers are charged a fixed base fare plus fare based on the distance travelled. During peak demand time, a surge fee also will be charged. Apart from this, if the customer books the vehicle in advance, advance booking fees will be charged. If the customer cancels the ride for some reason, a cancellation fee will be charged.

Exploration of problem statement

Input:

- Customer request comprising of location, destination and mode of travel
- Advance booking
- Cancellation of ride

Output:

- Suitable cab is assigned
- Generation of bill

Constraints:

- Driver is assigned based on
 - Minimum waiting time
 - Average rating
- Driver should be driving minimum distance

Exploration:

- Thorough research on the current available cab types, prices charged, local laws with regard to maximum fares, etc.
- Cab location is updated with each trip.
- Improving user experience by taking rating after the trip. The rating is updated in the repository.

Analysis using Data Flow Diagrams

DFD Level 0:

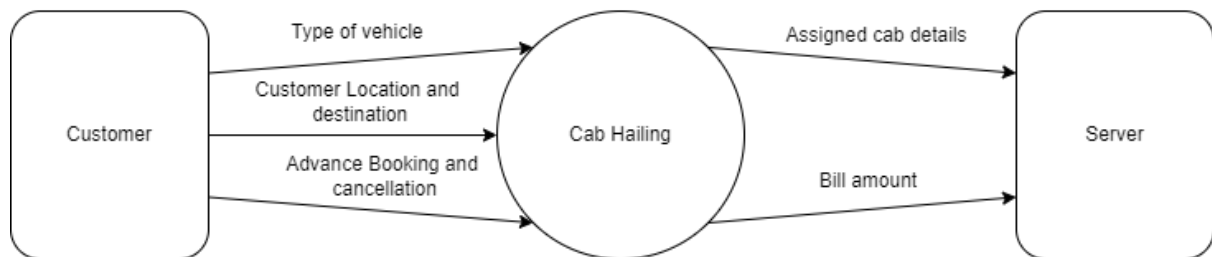


Figure 1: DFD L0

The software receives inputs like type of cab, customer location, customer destination, advance booking and cancellation information from the user. Upon receiving these inputs, it processes the data by combining with repository data and gives assigned cab details and final bill amount as the output.

DFD Level 1:

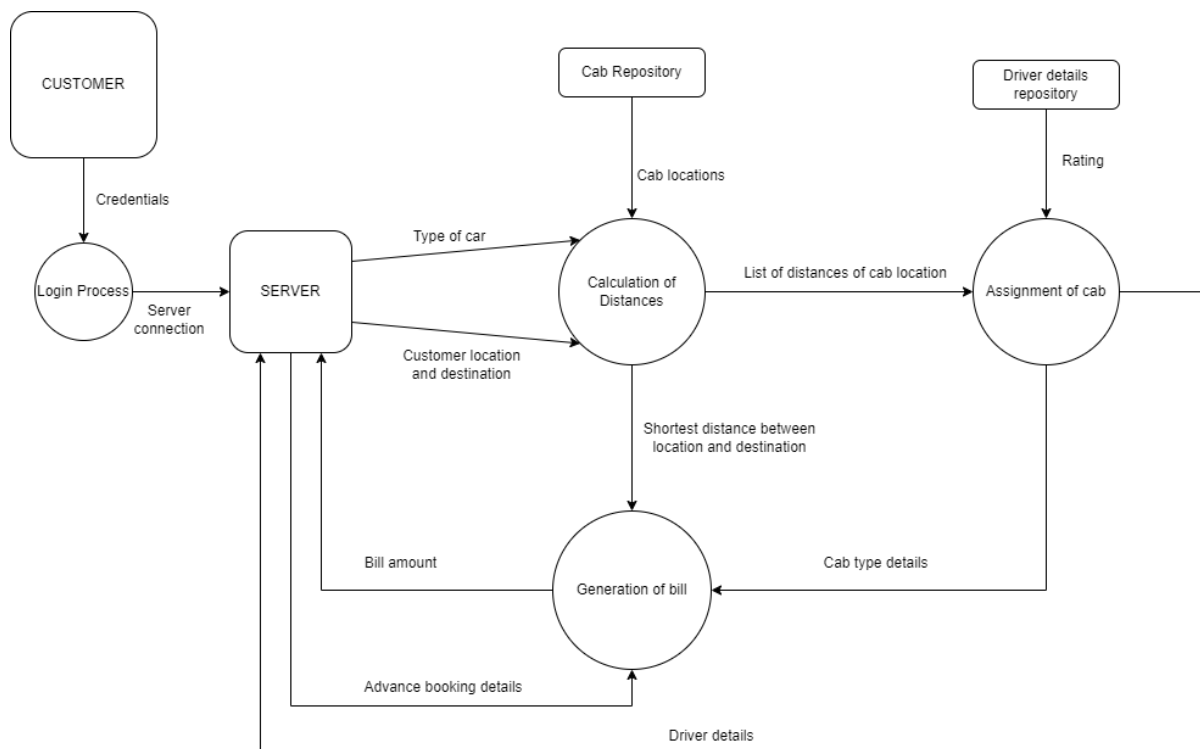


Figure 2: DFD L1

The software makes use of four different modules:

- Login process: Takes user credentials to login or register

- Calculation of distances: Gives array of distances between cabs and user location to 'Assignment of cab'. Also calculates shortest distance between customer location and destination and gives it to the process 'Generation of bill'.
- Assignment of cab: Calculates score between distance and driver rating to assign cab.
- Generation of bill: Takes all cost estimates to calculate bill.

DFD Level 2:

The level 2 DFD diagram goes deeper into the details of the processes at each stage of data flow. The sub processes that data has to undergo is clearly mentioned.

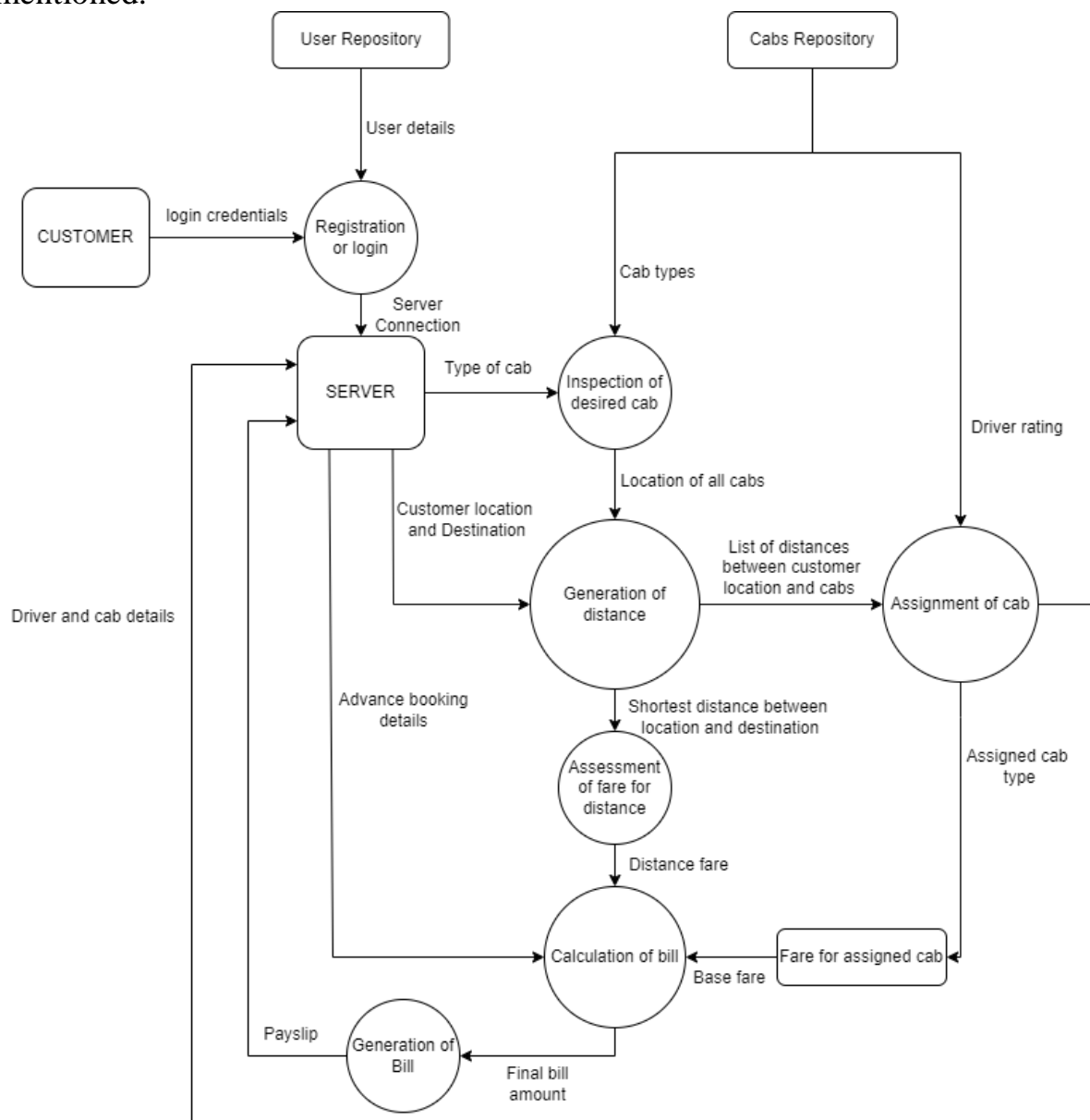


Figure 3: DFD L2

Detailed Design

SYSTEM ARCHITECTURAL DESIGN:

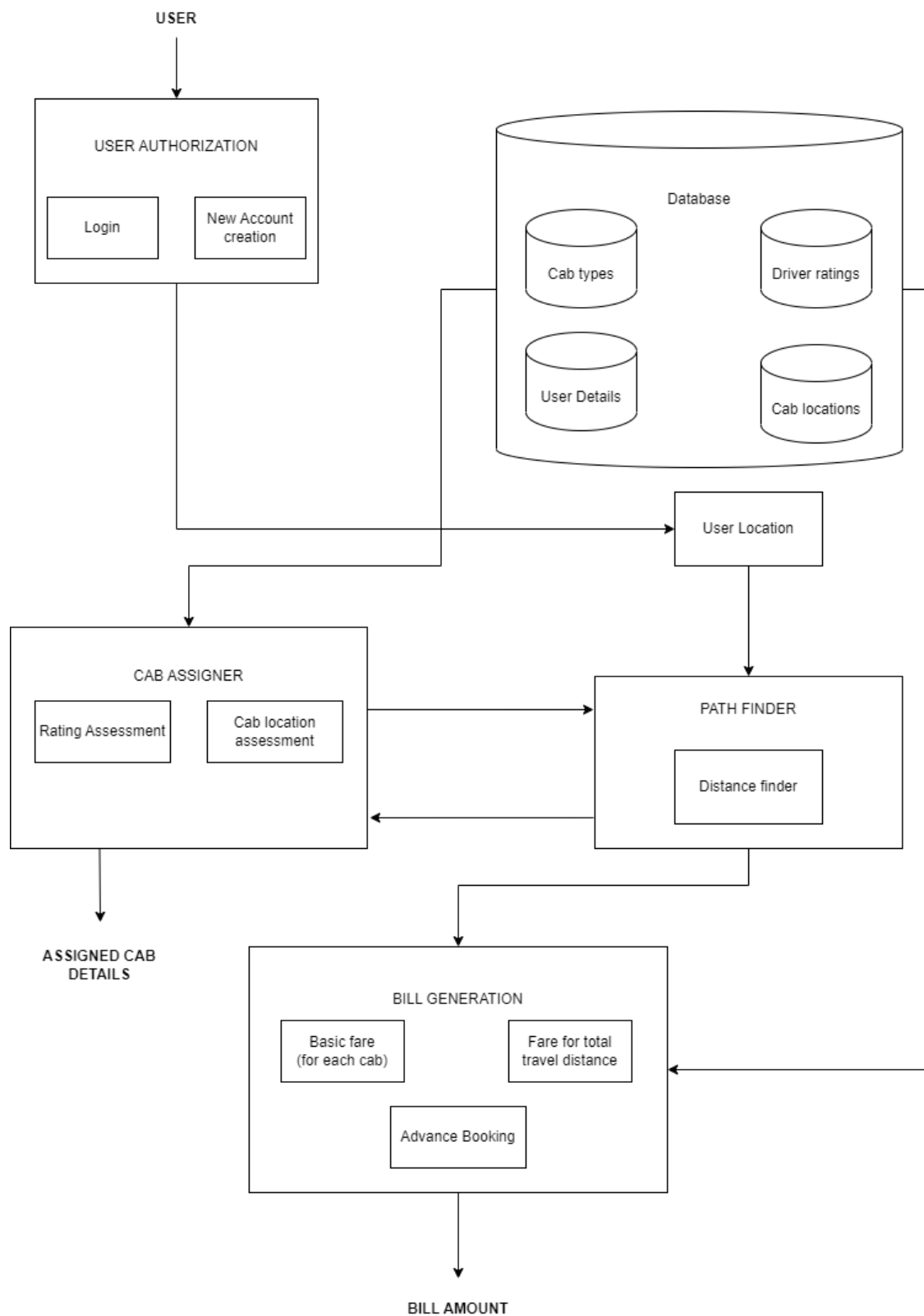


Figure 4: System Design

LOGIN MODULE:

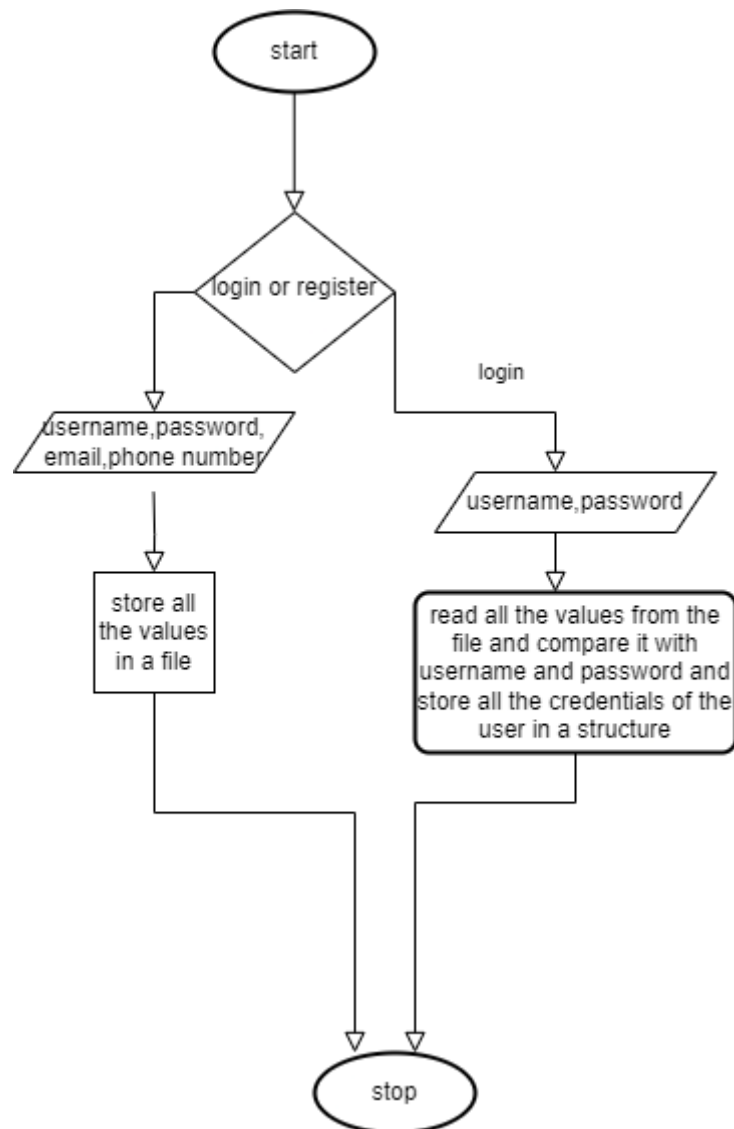


Figure 5: Login flowchart

Description:

It takes the user's input of whether they are trying to login or register and carries out the respective function.

If the user is trying to register, then details like name, email id, phone number, username and password are fetched from the user and are written into the user repository.

If the user wishes to login, then the username and password is fetched from the user and the details are compared with details in the user repository. If the right combination of username and password is given, then the application opens to the user server. Otherwise, it rejects the login action.

PATHFINDER MODULE:

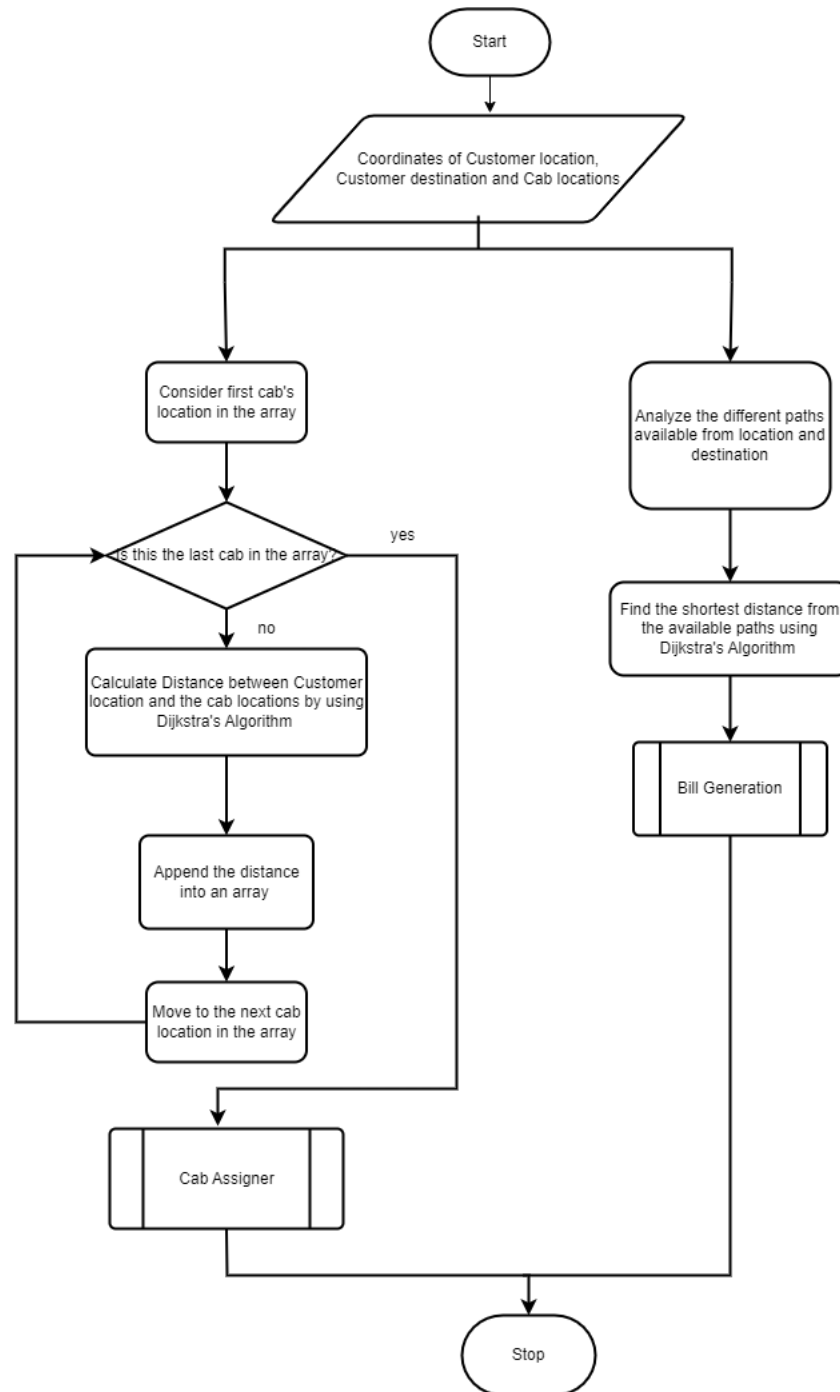


Figure 6: Pathfinder flowchart

Description:

The path finder module takes the cab locations, user location and destination as input data. The module uses A* search algorithm to find the distances.

The path finder produces an array of distances which are calculated by finding the difference between the user location and each of the desired cab locations. This array is then passed to the cab assigner module.

The same path finder is also used to find the path with the minimum distance from the user's current location to the final destination. This distance is passed to the bill generation module.

Our application gives the option of travelling within Chennai and also within Tamil Nadu. For locations within Chennai, we have used the A* search algorithm and for locations within Tamil Nadu, we have used the Dijkstra's algorithm. The Dijkstra's algorithm is used to find the shortest path from a single source node to all other nodes in a weighted graph with non-negative edge weights. The A* algorithm is an extension of Dijkstra's algorithm with added heuristics to efficiently find the shortest path between two points in a weighted graph or grid.

CAB ASSIGNER MODULE:

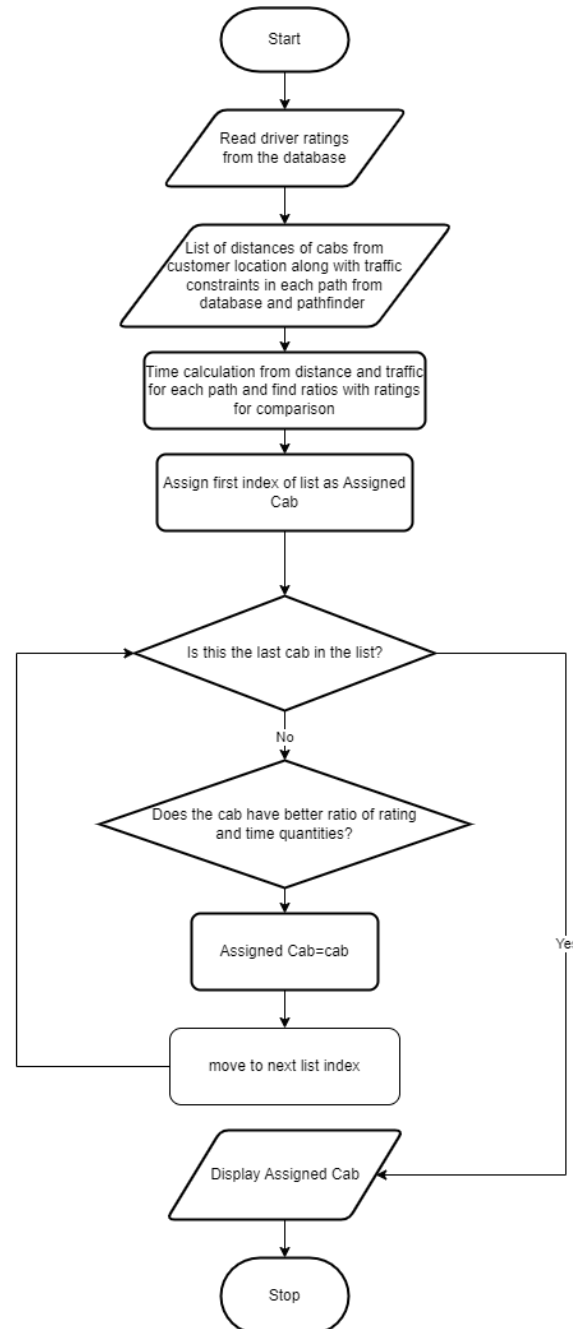


Figure 7: Cab assigner module

Description:

The cab assigner gets input of the array of distances between the user and each of the desired cabs, from the path finder module. These distances are then normalized using the formula:

$$normalised_distance = 1 - \frac{distance - min_distance}{max_distance - min_distance}$$

Then the weightage for normalized distance and rating is assigned. Here we have assigned distance_weightage = 70% and rating_weightage = 30%. After this the score for each cab is assigned using the formula:

$$score = (normalised_distance \times distance_weightage) + (rating \times rating_weightage)$$

After the scores are calculated, the cab with the highest score is assigned to the user. The cab details are then sent to the server.

BILL GENERATION MODULE:**Description:**

The module contains information about the base fare, fare based on distance, surge fee, advance booking fee and cancellation fee for each cab type. This module receives the shortest distance between the user location and destination as input. Also making use of the user's choice to book in advance or not, it calculates the bill. If the user wishes to cancel the trip, then the cab cancellation fee is added to the user's account and will be added to the bill of the next trip.

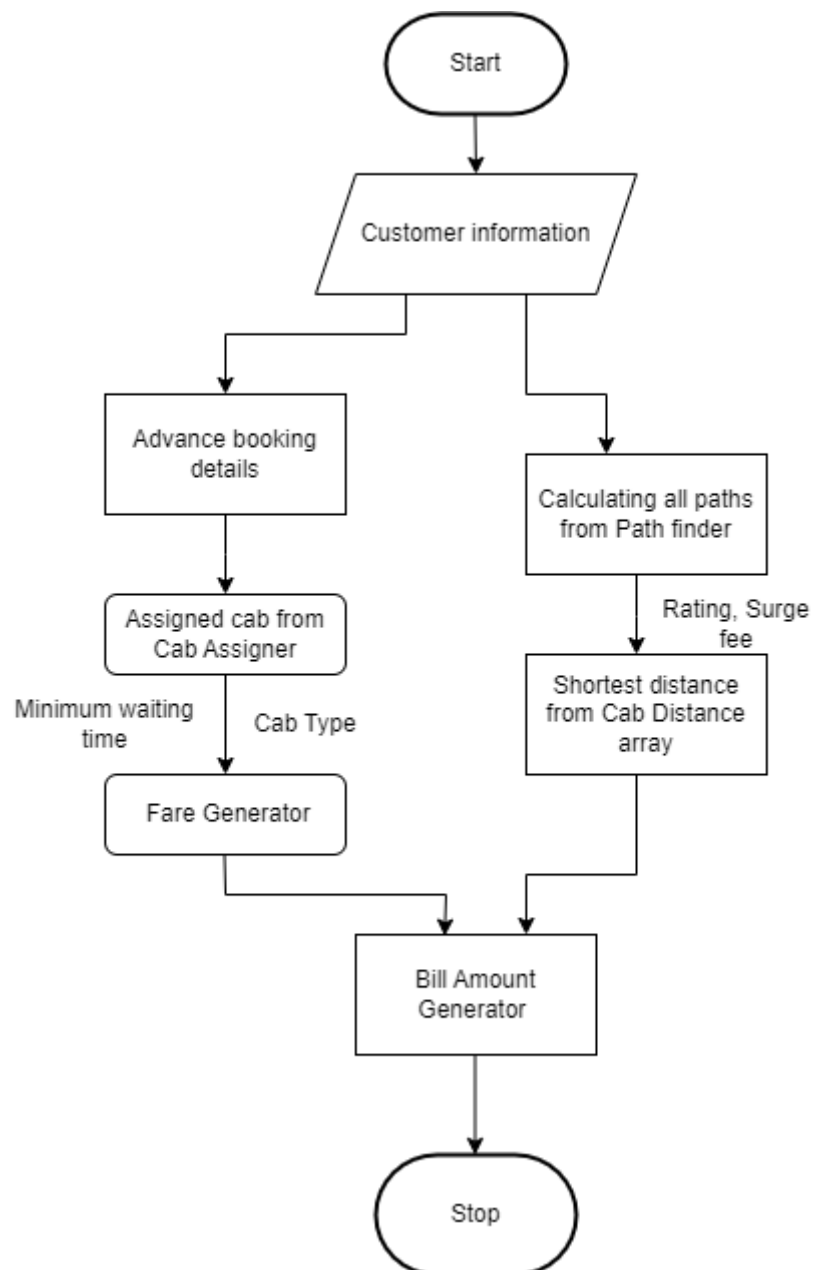


Figure 8: Bill generation module

Implementation

1. Data Organization

Login:

Since each user has their own set of name, phone number and the cancellation fee, which are of different data types, a structure is used.

Structure ("USER"):

- name of character data type array with 50 characters
- email of character data type array with 50 characters,
- phone number of character data type array with 50 characters
- username of character data type array with 50 characters
- password of character data type array with 50 characters
- cancellationfee of float data type.

As we have a lot of users, we can use array of structures to store it

File (“USERS.txt”): Here the array of users is directly written onto a file. We have used fwrite() function to easily write a structure in a file. fwrite() function writes to the file stream in the form of a binary data block. All components of structure “USER” is also present in the “USERS.txt” file.

Pathfinder:

A* search algorithm:

As all of the data here is of the same data type we can use arrays to store it.

Array (Grid): here arrays are used to store the actual map of the map we created, in the form of integers.

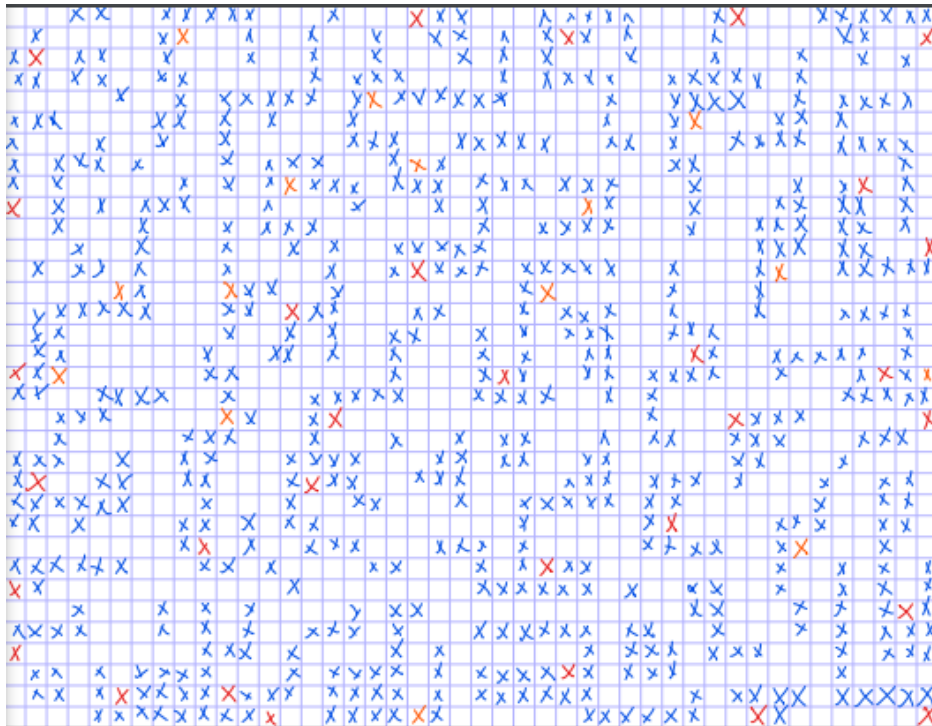


Figure 9: Grid used for A*

Array (places): It is also used to store the information about the coordinates of the map, in the form of a 2 dimensional array of integers.

Array (placenames): It is used to store the names of the locations in the form of strings.

16.nettukuppam	1.home	31.ripon building
17.thousand lights mosque	2.government museum	32.bm birla planetarium
18.fort St.george	3.grand gallada	33.anna centenary library
19.chetpet lake	4.arignar anna zoo	34.vgp snow kingdom
20.santhome church	5.jeeva park	35.vit college of engineering
21.phoenix mall	6.valluvar kottam	36.semmozhi poonga
22.kapaleeshvarar temple	7.ashtalakshmi temple	37.guindy national park
23.mgm dizzee world	8.connemara library	38.annai velankanni shrine
24.vdapalani murugan temple	9.iskcon temple	39.perignar anna memorial
25.t nagar	10.dakshikachitra museum	40.tholkappia poonga
26.parthasarathy temple	11.santhome cathedral	41.grand Square
27.vr mall	12.madras war cemetry	42.sivan park
28.cmbt bus stand	13.crocodile bank	43.ssn college of engineering
29.anna nagar tower	14.express avenue mall	44.dav public school
30.forum mall	15.shirdi sai baba temple	45.appolo hospital

Figure 10: All locations in the A* grid

Dijkstra's Algorithm:

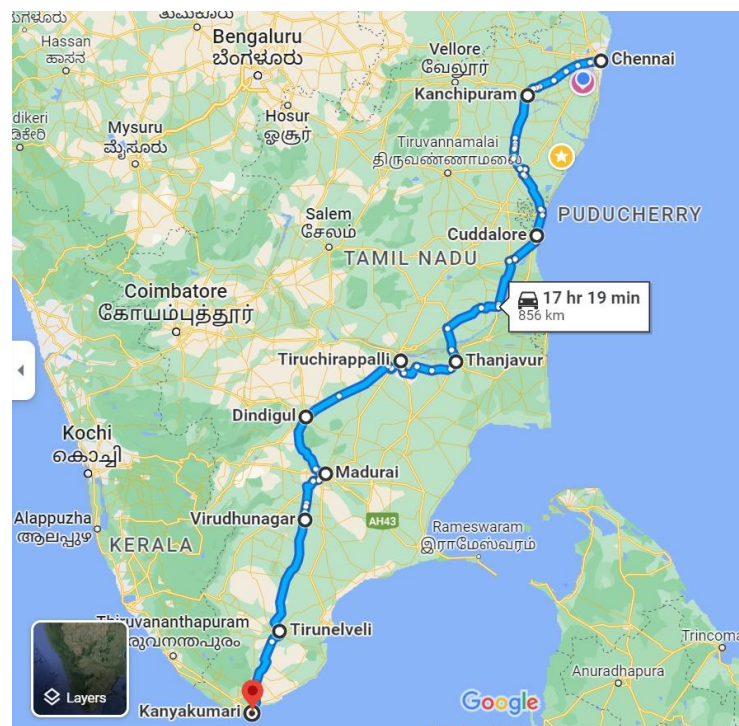


Figure 11: All nodes in the Dijkstra's algorithm

Array (distanceMatrix): 2-dimensional array representing the distance matrix between different cities. It stores the distances between cities in Tamil Nadu. The value at distanceMatrix[i][j] represents the distance from city i to city j.

Array(dist): 1-dimensional array of size NUM_CITIES (10 in this case) used in the Dijkstra's algorithm to store the shortest distances from the source city to all other cities. The value at dist[i] represents the shortest distance from the source city to city i.

1. Chennai
2. Kanchipuram
3. Cuddalore
4. Thanjavur
5. Tiruchi
6. Dindigul
7. Madurai
8. Virudhunagar
9. Tirunelveli
10. Kanyakumari

Figure 12: All locations in the Dijkstra's algorithm

Array(visited): 1-dimensional boolean array of size NUM_CITIES used in the Dijkstra's algorithm to keep track of visited cities. The value at visited[i] indicates whether city i has been visited or not. (Usage of boolean values for returning)

Array(parent): 1-dimensional array of size NUM_CITIES used in the Dijkstra's algorithm to store the parent node of each city in the shortest path. The value at parent[i] represents the parent of city i in the shortest path tree.

Cab assigner:

As all of the data here is of the same data type we can use arrays to store it.

Structure (driver)

- driver name of integer type
- cab type of character array type
- cab model of character array type
- register number of character array type
- rating of float type
- number_of_trips of integer type
- column number(in the A* grid) of integer type
- row number (in the A*grid) of integer type

File (“driver.txt”): has all the information about the driver and cab. It has the details about the driver separated by commas in the order of the structures. The details in the file is stored into an array of structure driver. All elements in the structure “driver” is also present in the file “driver.txt”.

Array (details): The distance and the rating are stored in the form of a 2d array and the cab score is calculated using this information.

Bill generator:

Array (cabtype): It has all the different cab types available.

{"minisedan", "SUV", "Sedan"}

Array (basefares): It has the base fares (in Rs.) of the respective cab types.

{75.61, 133.0, 45.40}

Array (fare_per_km): It has the fare per kilometer(in Rs.) values for each cab type.

{13.56, 25.60, 30.68}

Array (cab cancellation): It contains the cab cancellation charges (in Rs.) for each cab type found from analyzing market trends.

{15.80, 31.60, 23.70}

2. Explanation of any other libraries used

<stdio.h>	For all standard input and output functions
<stdlib.h>	For random functions and other utility functions
<time.h>	For date-time functions
<string.h>	For string functions like strcmp()
<stdbool.h>	For making use of bool as a Boolean data type
<math.h>	For basic math functions like sqrt()
<ctype.h>	For basic character type functions like tolower()
<unistd.h>	For symbol constants and types
<sys/ioctl.h>	For performing control functions

3. User interface design

We have not implemented a proper UI design or dynamic user interface however we have worked on console design to make the entire booking process easier to visualize and comfortable to use. We have also made the effort to provide concise and easy-to-understand questions to guide the user. The format type has also been specified in all cases to enhance the implementation of the code.

4. Platform used for Code Development

The platform used for code development is Repl.it. We chose to work with Repl.it as it provides real-time collaborative coding, allowing team members to work together simultaneously thus facilitating seamless collaboration and enhanced productivity. Being a cloud-based platform, repl.it also eliminates the need for local installations and configurations. Therefore we could access the project from any device with an internet connection, providing flexibility and ease of access.

Validation through Detailed Test cases

Test Case No.1: User Registration

```
Welcome to our cab hailing system!

Let's get you started!
What would you like to do?
1. Login
2. Register

Enter Choice: 2

Enter your name: Neha
Enter your Email ID: neha@gmail.com
Enter phone number: 9348782703
Enter a username: neha
Enter your password: neha

Enter your username: neha
Enter your password: neh
Wrong password entered. Please try again.
Welcome to our cab hailing system!

Let's get you started!
What would you like to do?
1. Login
2. Register

Enter Choice: 2

Enter your name: Mugil
Enter your Email ID: mugil@gmail.com
Enter phone number: 9037429014
Enter a username: mugil
Enter your password: mugil

Enter your username: mugil
Enter your password: mugil

Mugil, You have successfully logged into your account!
Continue booking with us!!
```

Test Case No.2: User Login

```

Welcome to our cab hailing system!

Let's get you started!
What would you like to do?
1. Login
2. Register

Enter Choice: 1

Enter your username: neha
Enter your password: neha

Neha, You have successfully logged into your account!
Continue booking with us!!

Welcome to our cab hailing system!

Let's get you started!
What would you like to do?
1. Login
2. Register

Enter Choice: 1

Enter your username: mug
Enter your password: mugil
The given username doesn't exist. Please try again.
```

Test Case No.3: Within Chennai

Test Case No. 3.1: For cab type Mini Sedan

Case: Advance booking is done and trip is not cancelled

```

Welcome to our cab hailing system!

Let's get you started!
What would you like to do?
1. Login
2. Register

Enter Choice: 1

Enter your username: nik
Enter your password: nik

nik, You have successfully logged into your account!
Continue booking with us!!

Choose the mode of travel:
1. Within Chennai
2. Within Tamil Nadu

Enter choice:1

Enter the pickup location: crocodile bank
Enter the destination: vr mall

Enter the type of cab: minisedan

Do you want to book your ride in advance?
Enter yes/no: yes
Enter the number of hours for pre-booking: 10

Advanced Booking accepted
```

```

Driver Details

Driver's name: Murugan
Car Model: figo
Registration number of the cab: TN025445
Total distance travelled in km: 21.00 km

BILL

Time: 8:00
Total Bill Amount = Rs. 674.68

Bill split-up

Base fare: Rs. 75.61
Fare per Kilometre: Rs. 13.56
Advance booking fee: Rs. 37.81

Do you wish to cancel your trip?
no

You have reached your destination!!

We hope you had a safe journey.Please enter your feedback rating below.
Enter the cab rating: 4.0

Thank you for booking with us!

```

Case: Advance booking is not done and trip is cancelled

```

Choose the mode of travel:
1. Within Chennai
2. Within Tamil Nadu

Enter choice:1

Enter the pickup location: forum mall
Enter the destination: valluvar kottam

Enter the type of cab: minisedan

Do you want to book your ride in advance?
Enter yes/no: no

Advanced Booking rejected

Do you wish to continue booking? yes

Driver Details

Driver's name: Paul
Car Model: tigor
Registration number of the cab: TN115427
Total distance travelled in km: 39.00 km

BILL

Time: 20:00
Total Bill Amount = Rs. 918.76

Bill split-up

Base fare: Rs. 75.61
Fare per Kilometre: Rs. 13.56
Surge fee: Rs. 37.81

Do you wish to cancel your trip?
yes

A cancellation fee of 292.30 will be added to your next trip!
Thank you for booking with us!

```

Score calculation example:

```
Cab no. 1
cab distance (from user) = 20.00
rating = 4.20
fscore = 1.79
Cab no. 2
cab distance (from user) = 49.00
rating = 3.90
fscore = 1.17
Cab no. 3
cab distance (from user) = 11.00
rating = 3.80
fscore = 1.84
Cab no. 4
cab distance (from user) = 36.00
rating = 4.00
fscore = 1.44
```

Test Case No.3.2: For cab type Sedan

Case: Trip is booked in advance and is cancelled later

```
Choose the mode of travel:
1. Within Chennai
2. Within Tamil Nadu

Enter choice:1

Enter the pickup location: ripon building
Enter the destination: home

Enter the type of cab: Sedan

Do you want to book your ride in advance?
Enter yes/no: yes
Enter the number of hours for pre-booking: 10

Advanced Booking accepted

Driver Details

Driver's name: Nirmal
Car Model: Accent
Registration number of the cab: TN129232
Total distance travelled in km: 45.00 km

BILL

Time: 23:00
Total Bill Amount = Rs. 1480.30

Bill split-up

Base fare: Rs. 45.40
Fare per Kilometre: Rs. 30.68
Advance booking fee: Rs. 22.70

Do you wish to cancel your trip?
yes

A cancellation fee of 55.30 will be added to your next trip!
Thank you for booking with us!
```

Score calculation example:

```
Cab no. 1
cab distance (from user) = 24.00
rating = 4.30
fscore = 1.99
Cab no. 2
cab distance (from user) = 24.00
rating = 4.10
fscore = 1.93
Cab no. 3
cab distance (from user) = 42.00
rating = 4.10
fscore = 1.23
Cab no. 4
cab distance (from user) = 25.00
rating = 4.50
fscore = 2.01
```

Test Case No.3.3: For cab type SUV

Case: No advance booking and no cancellation of trip

```
Choose the mode of travel:
1. Within Chennai
2. Within Tamil Nadu

Enter choice:1

Enter the pickup location: jeeva park
Enter the destination: t nagar

Enter the type of cab: SUV

Do you want to book your ride in advance?
Enter yes/no: no

Advanced Booking rejected

Do you wish to continue booking? yes
```

```
Driver Details

Driver's name: Akash
Car Model: Innova
Registration number of the cab: TN884454
Total distance travelled in km: 38.00 km

BILL

Time: 13:00
Total Bill Amount = Rs. 1421.80

Bill split-up

Base fare: Rs. 133.00
Fare per Kilometre: Rs. 25.60

Do you wish to cancel your trip?
no
    You have reached your destination!!

We hope you had a safe journey.Please enter your feedback rating below.
Enter the cab rating: 4.5
    Thank you for booking with us!
```

Test Case No.4: Within Tamil Nadu

Test Case No.4.1: For cab type Mini Sedan

Case: The trip is booked in advance and cancelled later

```

Welcome to our cab hailing system!

Let's get you started!
What would you like to do?
1. Login
2. Register

Enter Choice: 1

Enter your username: neha
Enter your password: neha

Neha, You have successfully logged into your account!
Continue booking with us!!

Choose the mode of travel:
1. Within Chennai
2. Within Tamil Nadu

Enter choice:2

Enter the pickup location: Chennai
Enter the drop location: Kanchipuram

Enter the type of cab: minisedan

Do you want to pre-book or advance book your ride?
Enter yes/no: yes
Enter the number of hours for pre-booking: 8

Advanced Booking accepted

Driver Details

Driver's name: Murugan
Car model: figo
Registration number of the cab:TN025445

BILL

Distance: 170 km
Time: 3:00
Total Bill Amount = Rs. 2418.61

Bill split-up

Base fare: Rs. 75.61
Fare per Kilometre: Rs. 13.56
Advance booking fee: Rs. 37.81

Do you wish to cancel your trip?
yes

A cancellation fee of 15.80 will be added to your next trip!
Thank you for booking with us!
```

Test Case No.4.2: For cab type Sedan

Case: No advance booking and no cancellation of trip

```
Choose the mode of travel:
1. Within Chennai
2. Within Tamil Nadu

Enter choice:2

Enter the pickup location: Kanchipuram
Enter the drop location: Madurai

Enter the type of cab: Sedan

Do you want to pre-book or advance book your ride?
Enter yes/no: no

Advanced Booking rejected

Do u want to continue? yes

Driver Details

Driver's name: Suraj
Car model: Verna
Registration number of the cab: TN229802

BILL

Distance: 360 km
Time: 9:00
Total Bill Amount = Rs. 11128.70

Bill split-up

Base fare: Rs. 45.40
Fare per Kilometre: Rs. 30.68
Surge fee: Rs. 22.70

Do you wish to cancel your trip?
no
You have reached your destination!!

We hope you had a safe journey.Please enter your feedback rating below.
Enter the cab rating: 3.9

Thank you for booking with us!
```

Score Calculation example:

```
Cab no. 1
cab distance (from user) = 67.00
rating = 4.30
fscore = 1.29
Cab no. 2
cab distance (from user) = 67.00
rating = 4.10
fscore = 1.23
Cab no. 3
cab distance (from user) = 53.00
rating = 4.10
fscore = 1.93
Cab no. 4
cab distance (from user) = 56.00
rating = 4.50
fscore = 1.90
```

Case: Advance booking is done and no cancellation of trip

Choose the mode of travel:

1. Within Chennai
2. Within Tamil Nadu

Enter choice:2

Enter the pickup location: Madurai

Enter the drop location: Dindigul

Enter the type of cab: Sedan

Do you want to pre-book or advance book your ride?

Enter yes/no: yes

Enter the number of hours for pre-booking: 5

Advanced Booking accepted

Driver Details

Driver's name: Suraj

Car model: Verna

Registration number of the cab: TN229802

BILL

Distance: 340 km

Time: 18:00

Total Bill Amount = Rs. 10499.30

Bill split-up

Base fare: Rs. 45.40

Fare per Kilometre: Rs. 30.68

Advance booking fee: Rs. 22.70

Do you wish to cancel your trip?

no

You have reached your destination!!

We hope you had a safe journey. Please enter your feedback rating below.

Enter the cab rating: 4.5

Thank you for booking with us!

Test Case No.4.3: For cab type SUV

Case: No advance booking is done but trip is cancelled

Choose the mode of travel:

1. Within Chennai
2. Within Tamil Nadu

Enter choice:2

Enter the pickup location: Tirunelveli

Enter the drop location: Kanyakumari

Enter the type of cab: SUV

Do you want to pre-book or advance book your ride?

Enter yes/no: no

Advanced Booking rejected

Do u want to continue? yes

Driver Details

Driver's name: Akash

Car model: Innova

Registration number of the cab:TN884454

BILL

Distance: 140 km

Time: 23:00

Total Bill Amount = Rs. 3717.00

Bill split-up

Base fare: Rs. 133.00

Fare per Kilometre: Rs. 25.60

Do you wish to cancel your trip?

yes

A cancellation fee of 31.60 will be added to your next trip!

Thank you for booking with us!

Score calculation example:

```
Cab no. 1  
cab distance (from user) = 67.00  
rating = 3.60  
fscore = 1.08  
Cab no. 2  
cab distance (from user) = 34.00  
rating = 4.10  
fscore = 1.93  
Cab no. 3  
cab distance (from user) = 67.00  
rating = 4.00  
fscore = 1.20  
Cab no. 4  
cab distance (from user) = 61.00  
rating = 3.50  
fscore = 1.18
```

Limitations of the solution provided

- **Scalability:** Currently the application runs on limited number of data samples and variation from these repository data is not accounted.
- **Service providers:** Currently the application can't be accessed for use by cab drivers. Hence dynamic cab assigning and better user experience is not assured.
- **Real-world scenarios:** The map is custom made and the application is not updated on real-world scenarios like traffic, road blocks, etc.
- **History:** Presently the application doesn't keep track of the user's history and hence can't provide optimized solutions.
- **User Experience:** The application is not equipped with an efficient and dynamic user-friendly interface. Apart from that, options like cab sharing and mid-trip destination updating is also not provided.
- **Security and Privacy:** The user details, driver and cab details should be protected using encryption techniques, regular data backups, etc.
- **Maintenance and support:** Maintenance of the software including bug fixing, customer support, updates, etc. should be taken into account.

Observations

These are some of the observations we have come across in the process of making the software. Some of these observations have also been taken into consideration as we produced our application for improving performance, stability and reliability. Our software when extended with these observations and concerns can be equipped to be used in real-time applications.

Societal Observations:

- The application must be user-friendly, intuitive and compatible in order to be viable from the business point of view.
- Safety and security of the user is enhanced by usage of features like driver ratings and reviews.
- Pricing transparency and competitive fares should be carefully disclosed for customer satisfaction and trust.
- Feedback and ratings can improve the service being provided by the software.
- Accessibility of the application for the disabled should be considered.
- Inclusivity with regard to gender, language spoken, etc. of the users.

Legal Observations:

- App should comply with local regulations which includes licensing, permits, vehicle inspections and certifications.
- Insurance policy and requirements should be checked.
- Driver requirements like age restrictions, driver's license, driving history checks, etc. should be ensured.
- Consumer protection laws, data privacy and intellectual property rights, anti-discrimination laws, etc. should be complied.
- Pricing regulations should be followed.

Environmental Observations:

- Options for carpooling and choice of electric vehicles should be provided.
- Sustainability and service optimization by collecting data from trips.
- Environmental certifications to improve credibility.
- Public transport integration to give sustainable travelling options to the users combined with the cab service.
- Congestion and traffic management by including cab sharing options.

Ethical Observations:

- Fair and unbiased driver allocation should be done without any discrimination.
- Ensuring privacy of data and having consent for data collection is important.
- Ensuring job security for the drivers.
- Responsible use of technology like AI, etc.
- Efficient customer support and dispute regulation.

Learning Outcomes

- We have understood and explored the complexities of real-time online cab hailing software services.
- We have analyzed and understood user requirements and we have learnt to create a user-centered design.
- We have done a thorough market analysis to present real-time cost estimates and analyze the trends to predict customer behavior to generate required test cases.
- We have learnt project management skills like setting milestones, allocating resources and adhering to a timeline.
- We learnt to improve our application based on the constructive feedbacks given during the reviews.
- There were also various challenges and roadblocks that had to be overcome with inputs from all team members.
- We also had to foster problem-solving skills.

References

- <https://www.youtube.com/watch?v=-L-WgKMFuhE>
- <https://www.uber.com/global/en/price-estimate/>
- <https://economictimes.indiatimes.com/news/politics-and-nation/all-taxis-to-charge-government-set-fares-after-aug-22-high-court/articleshow/53654050.cms?from=mdr>
- <https://www.numbeo.com/taxi-fare/in/Chennai>
- <https://neo4j.com/developer/graph-data-science/path-finding-graph-algorithms/>
- <https://www.gatevidyalay.com/a-algorithm-a-algorithm-example-in-ai/>
- <https://www.educative.io/answers/what-is-the-a-star-algorithm>
- <https://www.freecodecamp.org/news/dijkstras-shortest-path-algorithm-visual-introduction/>
- <https://www.scaler.com/topics/data-structures/dijkstra-algorithm/>
- <https://www.uber.com/in/en/>
- <https://easternpeak.com/blog/how-to-develop-a-taxi-booking-app-like-uber/>
- <https://medium.com/@narengowda/uber-system-design-8b2bc95e2cfe>
- <https://www.techrajput.com/2018/10/different-types-of-uber-rides-in-india.html>