

# Project Report

## **VIT Cab Sharing Application for Students** **CSE2004 – Database Management Systems**

Submitted by

**17BCE2000 – ANEESH CHOPRA**

**17BCE0972 – ISHAN ARORA**

**17BCE0985 – BHAVIKA ARORA**

**17BCE2184 – FIZA RASOOL**

Under the guidance of

**Prof. R.Sathyaraj**

Bachelor of Technology  
in  
Computer Science and Engineering



**VIT<sup>®</sup>**

**Vellore Institute of Technology**

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computing Science and Engineering

October 2018

## Contents

1. Abstract
2. Introduction
3. Literature Survey
4. Tools and methodologies
5. Experiment and Result
6. Conclusion
7. References

## **ABSTRACT**

As education thrives in India, more and more students take admissions into college and universities to complete their higher education. In order to accommodate such an extravagant number of students, most colleges and universities are located in a remote location or on the outskirts of metropolitan cities. Thus, many students tend to book a cab till their hometown or the nearest metropolitan area to board a flight if their hometown is far away. And many successful implementations of car-pooling services have been implemented via start-ups such as Uber, Ola etc by matching riders based on their destination and timing, but due to the remote locations of colleges, these services are unavailable to the college students. If the students are fortunate enough, they can share a ride together to the desired location, otherwise, they would have to travel alone and incur a huge cost.

Hence, in order to develop an application to cater to these students, we need to find an attribute to uniquely identify each student. We need to update the list of students whenever a new user enters and establish a form of communication between the riders. We need to retrieve relevant personal details about the user, to allow them to conveniently filter them through the available list of riders for the user to be able to find and form a suitable cab sharing group. Next, we need to make sure the searching is hassle free by already making it go through a primary filter where the available rider should first and foremost be travelling on the same date, destination and around the same time. Alongside, we have to combine the cab-booking procedure as well to make sure that the group formed isn't redundant and that they can quickly finalize their booking in a breeze without any additional efforts, this is made by acquiring details of drivers ready to work as freelancer drivers as well as details about their cars which the group can search through.

## INTRODUCTION

The VIT Cab Sharing Application is a GUI (Graphical User Interface) Application built on Python which simplifies the search for partners to travel with and reduces your expenses by working as a live phonebook which consists a database of students who are willing to pool and drivers who are ready to provide their service. In order for the user to find the perfect travel partner for the journey, we filter and sort the live phonebook list by looking at the pick-up time proposed, destination and other metrics. For the application to work hassle-free on every laptop, we chose SQLite as our choice of database as it is preinstalled in most of the laptops now and is fast for small databases. Eventually, we intend to make the process of finding partners and drivers smooth and fast through a well-connected database with necessary details. This report discusses the result of the work done in the development of 'VIT Cab Sharing Application for Students'. Python was used for front-end since it provides a good connecting facility between the database and also enables us to create an elegant GUI Application without using any other software. SQLite was used in the backend as it is important to save all the data related to the application and since it is preinstalled in many laptops and suitable for small databases similar to the one, we will be using.

## LITERATURE STUDY

**Database:** A database is a collection of organized information, maintained such that it can be easily accessed, managed, and updated. Technically, a database is an electronic framework that enables information to be easily fetched, controlled and refreshed. <sup>[1]</sup>

**Database Management System:** A system software used for creating, managing databases. It makes it possible for user to create, read, update and delete data from the database. The DBMS manage three important things: the data; the database engine which allows data to be accessed, locked and modified; and the database schema, which defines the database's logical structure. These three foundational elements help provide concurrency, security, data integrity, and uniform administration procedures.

A database schema is the skeleton structure that speaks to the consistent perspective of the whole database. It characterizes its elements and the relationship among them. It can be divided into Physical Database Schema which relates to the actual storage of data and its form of storage; and Logical Database Schema which defines all the logical constraints that need to be applied in the data stored. <sup>[2]</sup>

**RDBMS (Relational Database Management Systems):** An RDBMS is a type of DBMS with a row-based table structure that connects related data elements and includes functions that maintain the security, accuracy, integrity and consistency of the data. The RDBMS typically provides data dictionaries and metadata collections useful in data handling. These programmatically support well-defined data structures and relationships. Data storage management is a common capability of the RDBMS, and this has come to be defined by data objects that range from binary large object (blob) strings to stored procedures. Data objects like

this extend the scope of basic relational database operations and can be handled in a variety of ways in different RDBMS. <sup>[3]</sup>

## TOOLS AND METHODOLOGY

### Tools:

#### Hardware Requirement

A Laptop in which the source code of the project is stored

#### Software Requirement

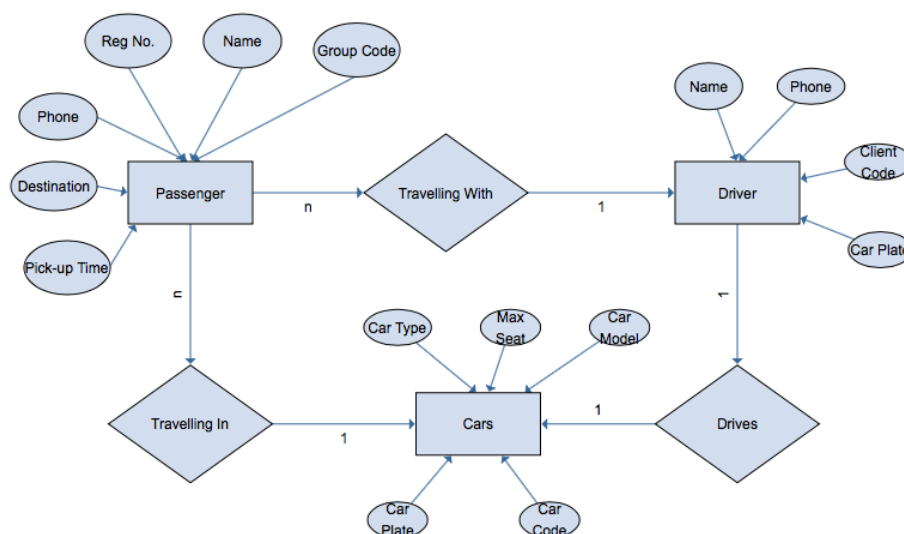
Front End: Python 3.0 (Free) [Libraries in use: Tkinter , sqlite]

Back End: SQLite (Free)

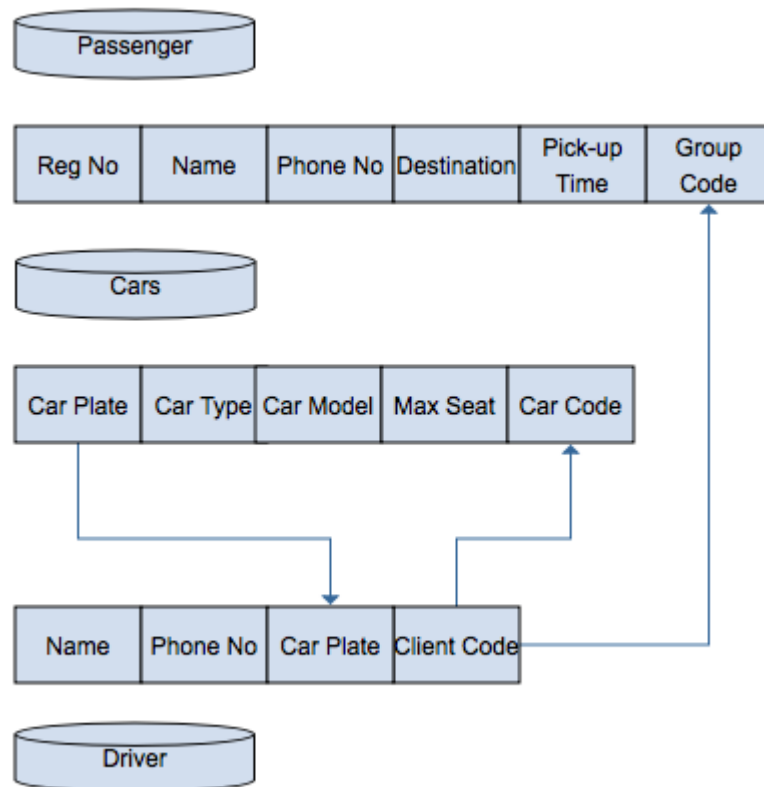
### Methodology:

1. Understanding the details required on students, cars, drivers and form a relational schema; Finally, Normalize the Database while trying keeping the number of tables at a bare minimum
2. Form a workflow and Define functions to perform required SQL Queries using sqlite library <sup>[4]</sup> <sup>[5]</sup>
3. Use Tkinter library to design and implement the GUI (Graphical User Interface) for the application <sup>[6]</sup>
4. Testing and debugging every aspect of the process of group-matching and car-booking after each and every update to the code, to ensure a safe and seamless experience

### ER DIAGRAM



## RELATIONAL SCHEMA



## EXPERIMENT AND RESULT

1) To check if a new user can understand and execute all processes in the application without any help from any other person

[Note: Instructions are provided in the Application itself to clarify all of the doubts that come to the user's mind]

### a) Signup/Login

Screen Shot 2018-08-03 00:38 AM Final DBMS REVIEW Adobe Photoshop CC 2018... [ckzSoft]

CabShare - VIT

**CABSHARE**

Personal Details					
Regno	17BEE0227	Name	Gaurav Mittal	Phone No	8836217303
		Gender	Male		

Trip Details			
Destination	chennai airport	Date(YYYY-MM-DD)	2018-11-18
Max Pickup Time (24-Hr Format)	21.3	GroupCode (Leave Alone If New)	

[Check Instructions For Help](#)

Review 1 - Image CAO

tk #2

If you are a old user, Simply enter your register number and click FIND RIDER  
If You Are New, Please Fill up all information besides the Groupcode;  
Click the "Add To List" Button;  
Then Click "Find Rider" To Move on with our Process

CabShare - VIT

**CABSHARE**

Personal Details					
Regno	17BEE0227	Name		Phone No	
		Gender			

Trip Details			
Destination		Date(YYYY-MM-DD)	
Max Pickup Time (24-Hr Format)		GroupCode (Leave Alone If New)	

[Check Instructions For Help](#)

**Result:** User was able to Sign up and Login very easily with the help of the instructions present in the application alone and required no further help

## B) Searching and Filters

CabShare - VIT

**SEARCHING FILTERS**

Register No.  Name

Phone  Gender

**LIVE PHONEBOOK (Reg No, Name, Phone No, Gender, Destination, Time)**

17BEE0227 {Gaurav Mittal} 9873493220 Male {	<input type="button" value="View all"/>
	<input type="button" value="Select Rider"/>
	<input type="button" value="Show Rider Group"/>
	<input type="button" value="Unselect Rider"/>
	<input type="button" value="Select Car"/>
	<input type="button" value="Find Later"/>

CabShare - VIT

**SEARCHING FILTERS**

Register No.  Name

Phone  Gender

**LIVE PHONEBOOK (Reg No, Name, Phone No, Gender, Destination, Time)**

227 {Gaurav Mittal} 9873493220 Male {chennai	<input type="button" value="View all"/>
432 {Aman Sharma} 9844143221 Male {chennai	<input type="button" value="Select Rider"/>
532 {Aman Sharma} 9844143221 Male {chennai	<input type="button" value="Show Rider Group"/>
532 {Yatin Sapra} 9844243221 Male {chennai air	<input type="button" value="Unselect Rider"/>
532 {Ishant Sapra} 9844243221 Male {chennai a	<input type="button" value="Select Car"/>
742 {Ishan Chandra Pradhan} 9844343211 Male	<input type="button" value="Find Later"/>
743 {Ishan Arora} 9844343211 Male {chennai ai	
921 {Sumit Anand} 8774553312 Male {chennai a	
921 {Shreyas Dholakia} 8774551112 Male {chen	
944 {Shikhar Dhawan} 9774551112 Male {chenn	

**Result:** User was able to use the searching filters efficiently to find himself as well as search for other male riders available to share a ride with on the particular date and time he provided.



### C) Group Formation

The screenshot shows a web application window titled "CabShare - VIT". The interface is divided into two main sections: "SEARCHING FILTERS" and "LIVE PHONEBOOK (Reg No, Name, Phone No, Gender, Destination, Time)".

**SEARCHING FILTERS**

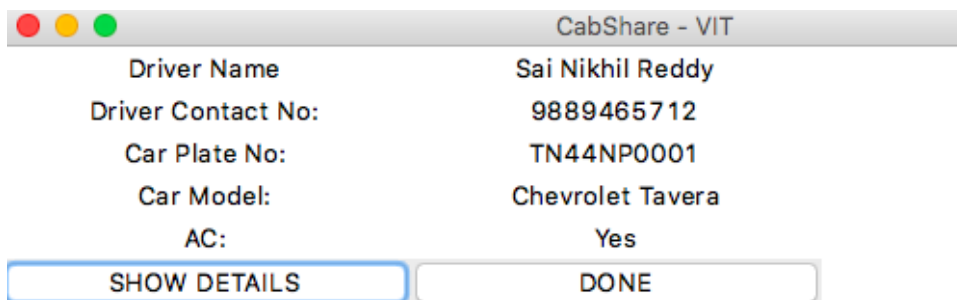
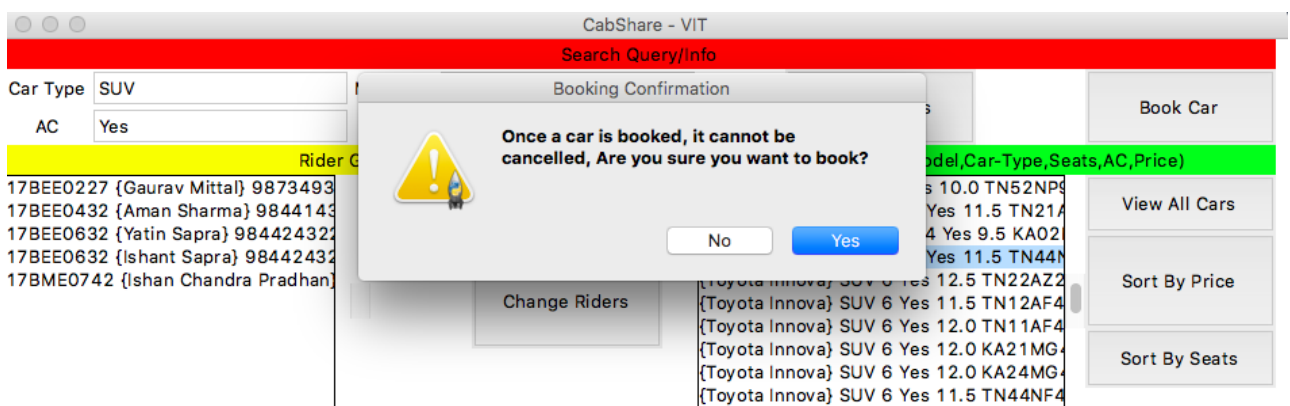
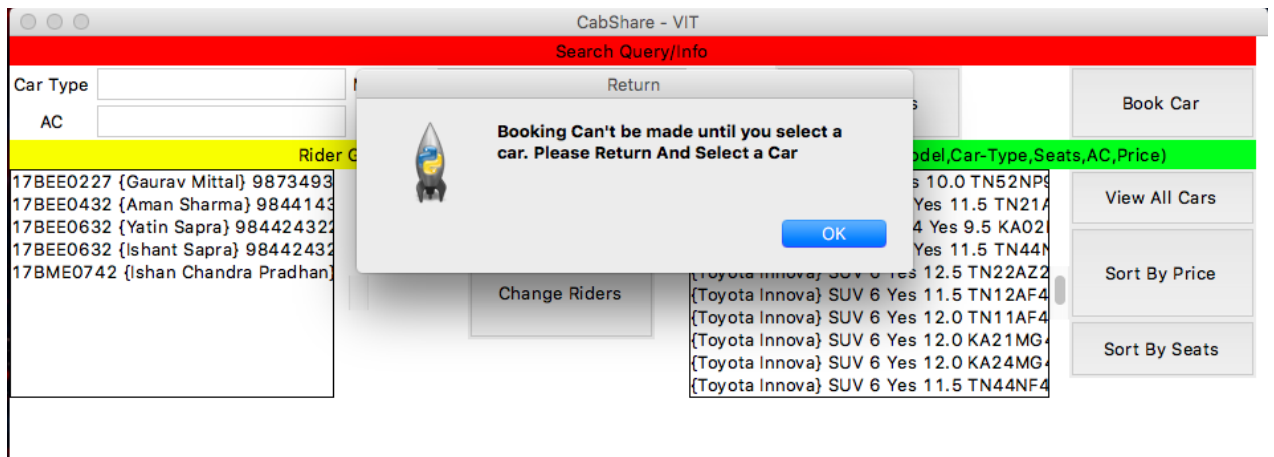
Register No.	17BME0742	Name	Ishan Chandra Pradhan	Search Riders
Phone	9844343211	Gender	Male	Search Same Year

**LIVE PHONEBOOK (Reg No, Name, Phone No, Gender, Destination, Time)**

17BEE0227 {Gaurav Mittal} 9873493220 Male {	View all
17BEE0432 {Aman Sharma} 9844143221 Male {	Select Rider
17BEE0632 {Yatin Sapra} 9844243221 Male {ch	Show Rider Group
17BEE0632 {Ishant Sapra} 9844243221 Male {c	Unselect Rider
17BME0742 {Ishan Chandra Pradhan} 98443432	Select Car
	Find Later

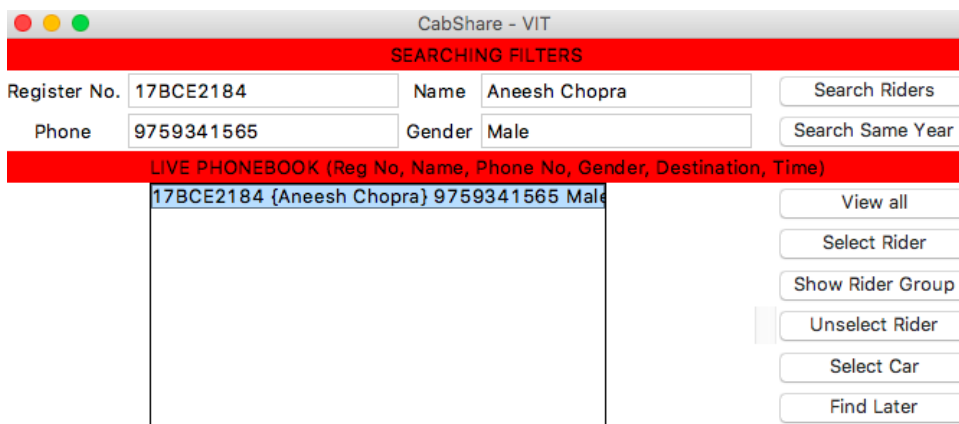
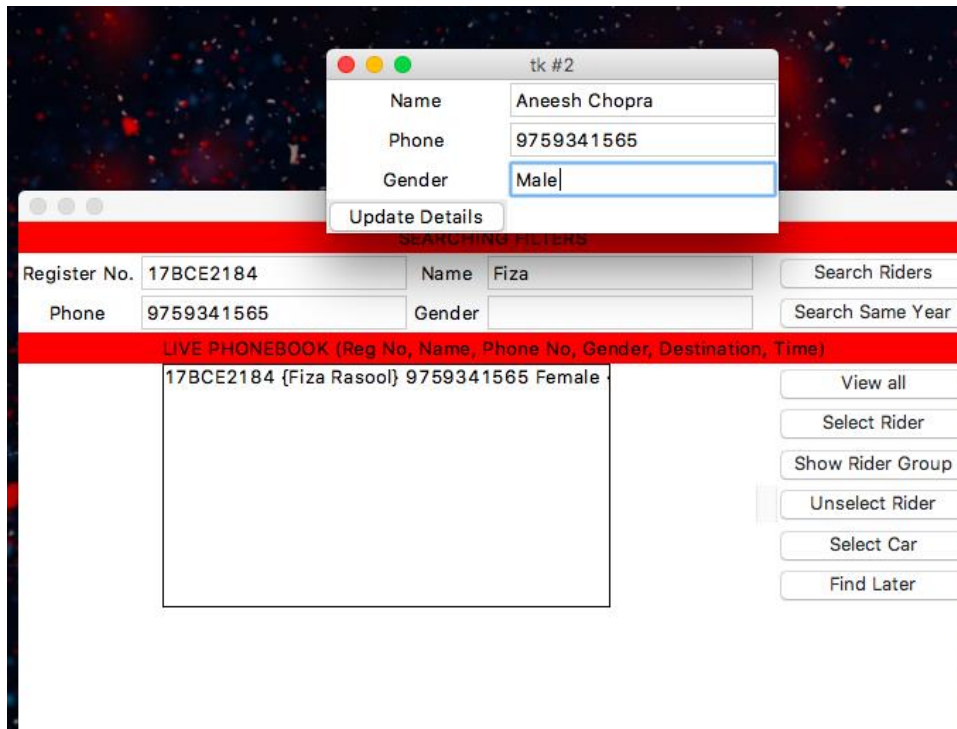
**Result:** User was able to Select/Unselect his riders without any confusion due to the self-explanatory buttons and was able to view his group in real-time to finalise his group and move on to the next step

## D) Car Selection/Car-Booking



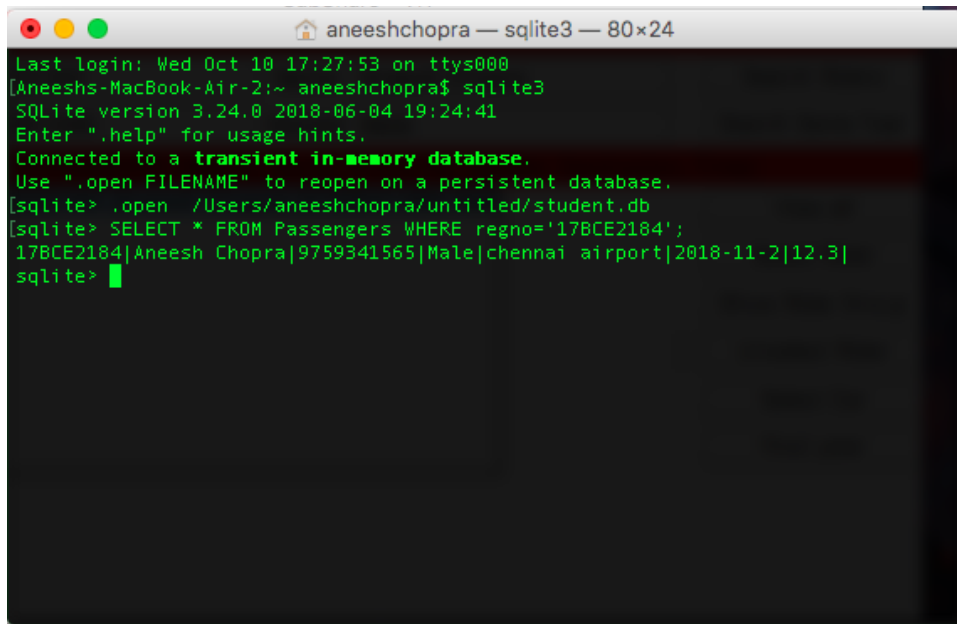
**Result:** The user was given a warning when he tried to book without selecting a car. When a car was selecting, he was given a confirmation pop-up message; Only on finalising the booking details, personal details about the driver were revealed to the user to establish contact

## E) Updation/Deletion



**Result:** User was able to simply change the details he wished to by using the 'Update Personal Details' available under the 'User' cascade in the Menu Bar.

## 2) Changes in Front-end Are Reflected in Back-end

A screenshot of a terminal window titled 'aneeshchopra — sqlite3 — 80x24'. The terminal shows the following text:

```
Last login: Wed Oct 10 17:27:53 on ttys000
[Aneeshs-MacBook-Air-2:~ aneeshchopra$ sqlite3
SQLite version 3.24.0 2018-06-04 19:24:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
[sqlite> .open /Users/aneeshchopra/untitled/student.db
[sqlite> SELECT * FROM Passengers WHERE regno='17BCE2184';
17BCE2184|Aneesh Chopra|9759341565|Male|chennai airport|2018-11-2|12.3|
sqlite>
```

**Result:** After updation, the changes were reflected on the back-end immediately with the help of the optimised sqlite library in Python.

### 3)Changes in Back-end are reflected in Front-end

```
[sqlite> SELECT * FROM Passengers WHERE regno='17BCE2184';  
17BCE2184|Aneesh Chopra|9759341565|Male|chennai airport|2018-11-2|12.3|  
[sqlite> UPDATE Passengers set name='Fiza Rasool' , gender='Female' WHERE regno='17BCE2184';  
[sqlite> SELECT * FROM Passengers WHERE regno='17BCE2184';  
17BCE2184|Fiza Rasool|9759341565|Female|chennai airport|2018-11-2|12.3|  
sqlite>
```

CabShare - VIT

**SEARCHING FILTERS**

Register No.	17BCE2184	Name	Fiza Rasool	Search Riders
Phone	9759341565	Gender	Female	Search Same Year

**LIVE PHONEBOOK (Reg No, Name, Phone No, Gender, Destination, Time)**

17BCE2184 (Fiza Rasool) 9759341565 Female	View all
	Select Rider
	Show Rider Group
	Unselect Rider
	Select Car
	Find Later

**Result:** Changes made in the Back-end were also immediately reflected in Front-End

## CONCLUSION

The Prototype proved to be a simple, better, time and cost-effective way to solve the problem of finding, connecting and matching college students who are travelling far distances on the same date together. With help of Normalization, we were able to get rid of redundant details we first thought of using and were able to make the whole process faster. Testing the application by letting random users helped us figure out what to include in the instructions which would significantly help the new users. Using Colour Theory, we understood the importance of the colour red, and used it wherever we thought useful information was to be communicated to the user. Letting college students, approach each other instead of auto-matchmaking helped avoiding machine error as well as allowing students to know each other better and finding a better rider as well. Including filters for searching through the list allowed the students, especially females, make safer choices and helped students get the best deal possible while choosing their car.

Throughout the project, we learned extensively about how to make complex queries in SQL, and get a glimpse of how it is used in the real world. Deciding to make a GUI, helped us in expanding our knowledge of Python and its libraries as well as how to establish a quick and strong connection with the database.

## REFERENCES

- [1] <https://searchsqlserver.techtarget.com/definition/database-management-system>
- [2] [https://www.tutorialspoint.com/dbms/dbms\\_data\\_schemas.htm](https://www.tutorialspoint.com/dbms/dbms_data_schemas.htm)
- [3] Pratt, Philip J.; Last, Mary Z. (2014-09-08). *Concepts of Database Management (8 ed.). Course Technology*. p. 29. ISBN 9781285427102.
- [4] [https://sebastianraschka.com/Articles/2014\\_sqlite\\_in\\_python\\_tutorial.html](https://sebastianraschka.com/Articles/2014_sqlite_in_python_tutorial.html)
- [5] <https://www.tutorialspoint.com/sqlite>
- [6] <https://docs.python.org/2/library/tkinter.htm>

