**CAB ALLOCATION SYSTEM-USING LINKED LIST**

**PROJECT REPORT**

Submitted for the course: Data Structures and Algorithms (CSE 2003)

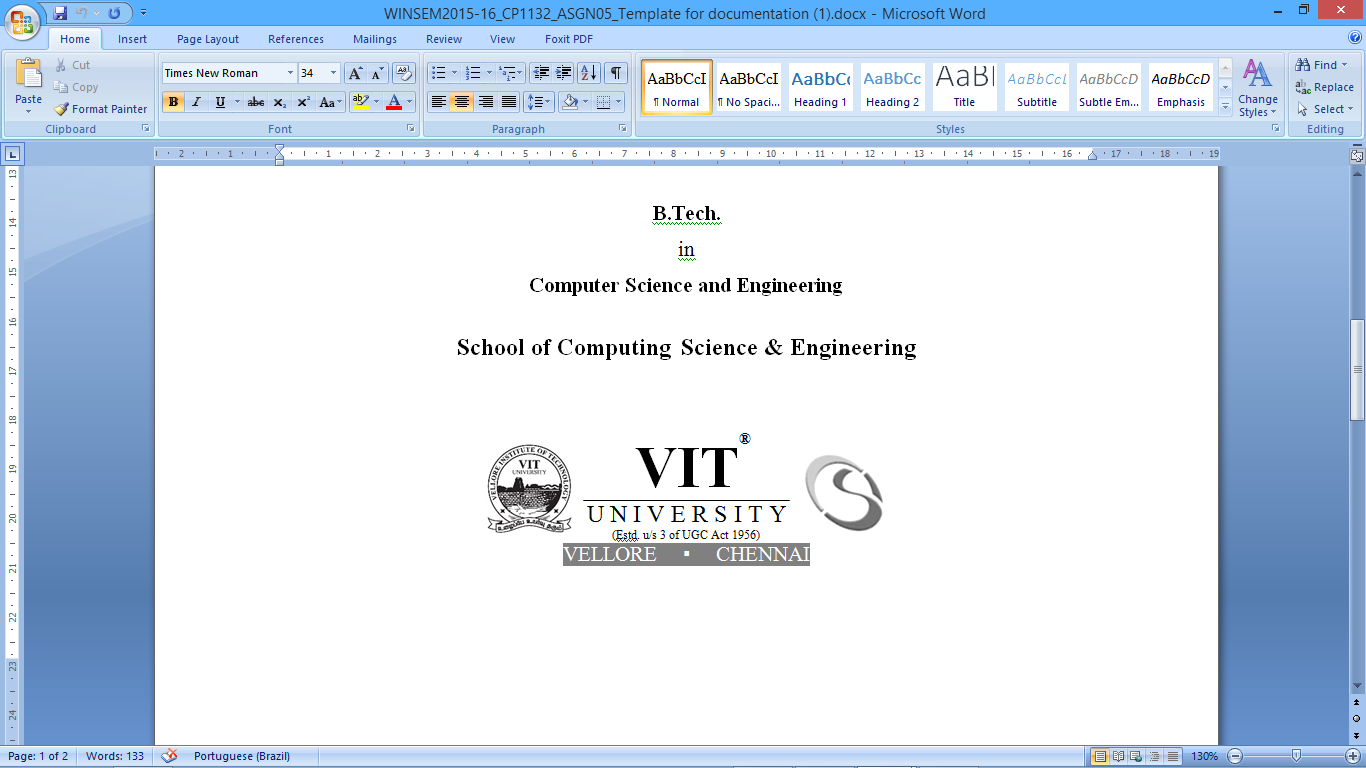
By

**(Name of students with reg. number)**

|  |  |
| --- | --- |
| **MAYANK VERMA** | **15BCE0032** |
| **VISHAL BHASKAR** | **15BCE0048** |

**Name of faculty: Prof. MURALI S**

**(SCHOOL OF COMPUTING SCIENCES AND ENGINEERING)**

****

**MAY,2016**

**CERTIFICATE**

This is to certify that the project work entitled “cab allocation system-using linked list” that is being submitted by “Mayank Verma and Vishal Bhaskar” for Data Structures and Algorithms (CSE 2003) is a record of bonafide work done under my supervision. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted for any other CAL course.

Place :Vellore

Date:

**Signature of students:**

**(Mayank) (Vishal)**

**Signature of faculty:**

**(Prof. MURALI S)**

**ACKNOWLEDGEMENTS**

We, group members would like to acknowledge our teacher Mr. Murali S for giving us the required inputs in form of thoughts and speech so as to understand the subject better. Furthermore, we would thank VIT University for being a supporter for the new projects. It could not be done without the whole team getting involved in the topic with their heart and soul.

**(Mayank) (Vishal)**

**ABSTRACT**

In this project we want to make a data management system for traveler association using doubly linked list. Here the user will be allowed to login as a customer or server (driver). Then, as a customer he can check the availability of a vehicle and thus book it. As a server he can give information to database when he is booked (and the place to which he is going) or when he reaches back after finishing a task. Therefore, we would be able to solve the problem of non availability and improper management of car. We tried to do something new, to show a good alternative.

1. **Introduction:**
   1. **Objective and goal of project:**

This project aims at presenting a new alternative for data management for cab allocation system. In this project we use linked list to store the data and to change it as required. We create linked lists which are linked in both downward and sideways. There are 3 user interfaces present here. One for the admin who can manipulate the data of the system. Another for the driver which can provide the information before leaving and coming back. And the last one is for the user to book the cab of desired type.

* 1. **What is Linked-list??**

In computer science, a linked list is a linear collection of data elements, called nodes pointing to the next node by means of a pointer. It is a data structure consisting of a group of nodes which together represent a sequence. Under the simplest form, each node is composed of data and a reference (in other words, a link) to the next node in the sequence.

1. **Traversal**

**2.1 Type of linked list used**

Here we used a linked list which has one main column in which all data pieces are linked in both downward direction and sideways direction. In the data element which are linked to the main sideways we linked it to the next sideways only.

**2.2 The downward traversal**

In the main column we used the data elements to store the cab types. The different types of cab that are available are stored in it. The admin can add more and delete from it. If a new whole cab type is introduced, it is linked to the downside chain. When a user enters a particular cab type, we traverse only downside to find the matching type.

**2.3 The sideways traversal**

In the side elements we store the different cab number which are available under a particular cab type and link them to the cab type list element. We are also storing the availability of the particular cab number in it. Whenever a particular cab type is entered by the user we first search for the cab type downside and when the cab type is found we start searching for a cab which is available that is its status is 1. If a cab is found we choose it and change its status to 0 when the user books it.

1. **The User interfaces**

**3.1 The admin interface**

Here one can manipulate the data in the allocation system. It is thus password protected for the safety purposes. The admin can introduce a new cab type or just introduce a new cab in existing cab type. He can delete the whole cab type or a particular cab, like when the cab is out of service, he can delete it. The admin can also view all the cab type present and their status.

**3.2 The driver interface**

This interface is for the drivers of the cabs. Here the driver is asked for his cab number, then we traverse in the linked list to find the cab. Then he is asked if he is coming back or leaving. If he is leaving, the status of his cab changes to 0 and if he is coming back, the status of his cab changes from 0 to 1. Thus helping in data management.

**3.3 The customer interface**

This is the interface which will be available for the public. Here the user is asked for a cab type he wants to book. Then we search for the cab type in the linked list web. After the particular type is found, we traverse sideways to find if any cab under the particular cab type is available or not. If available, we ask for booking confirmation and change the status of the cab to 0.

**Conclusion:**

Here we tried to store the data in an efficient way to reduce the time complexity and thus reduce the cost. We linked them both sideways and downside. It easily searches for cab type and then only it goes for searching the particular cab number which is available. We don’t need to cover all the data to search for a single cab, thus saving the time. We made different interfaces to make it easy to manipulate the entries, to maintain the allocation system, where the driver also contributes for the maintenance of the data. We tried to make the data management system more efficient than previous.