



**PES UNIVERSITY, Bangalore**  
(Established under Karnataka Act No. 16 of 2013)

**UE18CS202**

**B.Tech, Sem III**  
**Session : Aug-Dec, 2019**

**UE18CS202 – Data Structures**

**REPORT**  
**ON**  
**“Project Title”**

**SECTION :**

#	SRN	Name	Contact No.	Email ID	Sign
1.	PES1201800368	Pranavi	70228 50555	letters.to.pranavi@gmail.com	
2.	PES1201800369	Swanuja	99726 84218	swanuja2000@gmail.com	

## **ABSTRACT**

Brief introduction about the Project, the data structure used, the prototypes described and functionality implemented needs to be described in this section. The Advantages and use cases should be explained. (Minimum 8 lines)

The project is a simulation of a ticket reservation system. The data structure used is linked list implementation of queues.

The function prototypes are:

```
NODE* create_newNODE()
QUEUE* createQUEUE()
void enQUEUE(QUEUE* q, NODE* person)
NODE deQUEUE(QUEUE* q)
```

The advantages of this system is that when a situation arises such that all the seats in a train are booked, the next set of people who try to reserve a seat are automatically added to a queue and when a booked seat gets cancelled, the first person in the queue is automatically assigned that seat. The logic of this system can also be used for other ticketing systems, token counter systems, etc.

While executing the program, we can observe the reservation of a seat, enqueue and dequeue functions, cancellation of a seat, assignment of the cancelled seat and viewing of booking details.

## **DESCRIPTION OF DATA STRUCTURE, LOGIC AND FUNCTIONALITY**

Write a detailed procedure adopted like logic, functionality implemented in your project in detail. (Minimum 1 page)

The data structure used in this project is queues and we have adopted the linked list implementation of queues. Our seating arrangement is shown by an array of structures. Our main function has a switch case to choose between reserving a seat, cancelling a seat and viewing booking details.

A struct NODE is created which stores the person's details and other important details such as booking status, reservation number, waiting number, etc. A new NODE is created any time a person wants to reserve a seat using create\_newNODE(). The code then iterates through the array to search for an empty seat. If found, it assigns it to the person and their reservation number is given in case they need to check their details in the future. If no empty seat is found, a queue is created and the person is automatically added to the queue using the enQUEUE function and is given a waiting number to check their booking status in the future.

When a person wants to cancel their seat, they need to provide their reservation number and their seat is cancelled. The queue then gets dequeued using the deQUEUE function and the dequeued person is automatically assigned the cancelled status and their booking status in their struct NODE is changed accordingly.

When a person wants to view their details, they need to enter their reservation number/waiting number. If the person has a seat confirmed, a message saying the same is printed. If the person is still in the queue, a message stating the same is printed and requests the person to check in later again.

## **CONCLUSION**

What have you learnt from this project and how you can improve the concept can be discussed in this part of the report. (Minimum 6 lines)

Through this project, we learnt the real life application of data structures such as queues and linked lists and how helpful they are in solving problems. While coding, we came across quite a few logical errors and learned how to solve it such as error handling, using switches, memory allocation, etc. which we learnt to deal with.

We can improve this project in the future by incorporating file handling which permanently stores the user data and booking status.