

ABSTRACT

File structures provide efficient access to disk by allowing data to be stored, located and retrieved in a convenient way. A file system must be able to store the file, locate the file and retrieve the file. Hotel Management system is a file structure system implemented using Inverted lists. After adding records into the file operations like reading the details, searching the record, deleting the record and modifying a record can be performed on the contents of the file.

This mini project is aimed at providing a common platform for all kinds of hotel. Owing to the need to have all the customer details stored securely, facilitating timely access to update, search and delete, the program was built. Each user is identified by his room number and all the room details are stored in files. The program is built using C++ and Inverted list indexing concept.

It uses Inverted list in which a key may be associated with a list of reference fields pointing to documents that contain key. Files such as our secondary key leads to a set of one or more primary keys are called inverted lists. It is called inverted list because we are working our way backward from a secondary key to primary key to the record itself.

ACKNOWLEDGEMENT

The fulfilment and rapture that go with the fruitful finishing of any assignment would be inadequate without the specifying the people who made it conceivable, whose steady direction and support delegated the endeavours with success.

We would like to profoundly thank **Management of RNS Institute of Technology** for providing such a healthy environment to carry out this Project work.

We would like to thank our beloved Director **Dr. H N Shivashankar** for his confidence filling words and support for providing facilities throughout the course.

We would like to express our thanks to our Principal **Dr. M K Venkatesha** for his support and for inspiring us towards the attainment of knowledge.

We wish to place on record our words of gratitude to **Dr. M V Sudhamani**, Professor and Head of the Department, Information Science and Engineering, for being the enzyme and master mind behind our Project work.

We would like to express our profound and cordial gratitude to our Coordinator **Mr. Santhosh Kumar**, Assistant Professor, Department of Information Science and Engineering for her valuable guidance, constructive comments and continuous encouragement throughout the Project work.

We would like to express our profound and cordial gratitude to our Faculty Incharge **Mr. T S Bhagavath Singh**, Associate Professor, Information Science and Engineering, for his valuable guidance in preparing Project report.

We would like to thank all other teaching and non-teaching staff of Information Science & Engineering who have directly or indirectly helped us to carry out the project work.

And lastly, we would hereby acknowledge and thank our parents who have been a source of inspiration and also instrumental in carrying out this Project work.

ARYAN PANDEY

1RN16IS018

ASHISH SINGH

1RN16IS020

Table of Contents

Certificate	
Abstract	i
Acknowledgment	ii
Table of Contents	iii
List of Figures	v
List of Tables	vi
1 Introduction	1
1.1 History	1
1.2 About the File	2
1.3 Various Kinds of storage of Fields and Records	2
1.4 Application of File Structure	6
2 System Analysis	7
2.1 Analysis of Application	7
2.2 Structure used to Store the Fields and Records	7
2.3 Operations Performed on a File	8
2.4 Indexing Used	9
3 System Design	10
3.1 Design of the Fields and records	10
3.2 User Interface	10
3.2.1 Insertion of a Record	10
3.2.2 Deletion of Record	10
3.2.3 Display of Records	11
3.2.4 Searching of records	11
3.2.5 Modifying of Records	11
4 Implementation	12
4.1 About C++	12
4.1.1 Classes and Objects	12

4.1.2	Memory allocation and pointers	12
4.1.3	File Handling	12
4.1.4	Character Arrays and Character functions	13
4.2	Pseudocode	13
4.2.1	Reading Customer Details Module Pseudocode	13
4.2.2	Display Module Pseudocode	14
4.2.3	Deletion Module pseudocode	15
4.2.4	Search module pseudocode	15
4.2.5	Update module pseudocode	16
4.3	Testing	17
4.3.1	Unit Testing	17
4.3.2	Functional Testing	17
4.3.3	Integration Testing	19
4.3.4	System Testing	20
4.3.5	Acceptance Testing	22
4.4	Discussion of Results	23
4.4.1	Menu options	23
4.4.2	Insertion	24
4.4.3	Updation	25
4.4.4	Deletion	25
4.4.5	Display	26
4.4.6	Searching	26
4.4.7	File Contents	27
5	Conclusion and Future Enhancements	30
	References	31

List of Figures

Figure 1.1	Four methods for field structures	04
Figure 1.2	Making Records Predictable number of Bytes and Field	05
Figure 1.3	Using Length Indicator, Index and Record Delimiters	06
Figure 4.1	User Menu Screen	23
Figure 4.2	Hotel Record Implementation	23
Figure 4.3	Indexing of records based on secondary index	24
Figure 4.4	Inserting Customer Record	24
Figure 4.5	Modifying Customer record	25
Figure 4.6	Deleting Customer Record	25
Figure 4.7	Snapshot of display of records	26
Figure 4.8	Snapshot of displaying of the inverted index	26
Figure 4.9	Snapshot of displaying of inverted list with primary key	27
Figure 4.10	File contents containing record	27
Figure 4.11	Primary file containing room number and their position	28
Figure 4.12	Secondary file 1 containing name and their position	28
Figure 4.13	Secondary file 2 containing name and their room number	29

List of Tables

Table 4.1 Unit Test Cases for Hotel record details	17
Table 4.2 Functional Test Cases for Room No	18
Table 4.3 Integration testing for all modules	19
Table 4.4 System Testing for Hotel System	21
Table 4.5 Acceptance Testing for Hotel record details	22

