**Visvesvaraya Technological University**

**Belagavi-590018, Karnataka**



A Mini Project Report on

**“INDEXING FOR BANK DATA”**

**Submitted in partial fulfilment of the requirement for the**

**VI semester File Structures Laboratory [17ISL68]**

**Bachelor of Engineering**

**in**

**Information Science and Engineering**

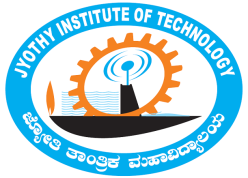
**Submitted by:**

**JAHNAVI S[1JT17IS015]**

**Guided by:**

**Mr. VADIRAJA A**

**Asst. Professor, dept of ISE**



**Department of Information Science and Engineering**

**Jyothy Institute of Technology**

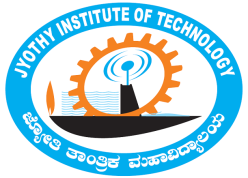
**Tataguni, Bengaluru-560082**

**2019-2020**

**Jyothy Institute of Technology**

**Tataguni, Bengaluru-560082**

**Department of Information Science and Engineering**



**CERTIFICATE**

Certified that the mini project work entitled **“indexing for library data”** carried out by **JahnaviS [1JT17IS015]** bonafide student of **Jyothy Institute of Technology**, in partial fulfilment for the award of **Bachelor of Engineering** in **Information Science and Engineering** department of the **Visvesvaraya Technological University, Belagavi** during the year **2019-2020**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the said Degree.

**Mr.Vadiraja A Dr.Harshavardhan Tiwari**

Guide, Asst . Professor Professor and HoD

Dept. Of ISE Dept. Of ISE

External Viva Examiner Signature with Date :

**ACKNOWLEDGEMENT**

Firstly, I am very grateful to this esteemed institution **“Jyothy Institute of Technology**” for providing me an opportunity to complete our project.

I express my sincere thanks to our Principal **Dr.Gopalakrishna K for** providing us with adequate facilities to undertake this project.

I would like to thank **Dr.HarshwardhanTiwari, Associate Professor and Head** of **Information Science and Engineering** Department for providing for his valuable support.

I would like to thank our guide **Mr.Vadiraja A, Asst. Prof.** for his keen interest and guidance in preparing this work.

Finally, I would thank all my friends who have helped me directly or indirectly in this project.

**JAHNAVI S**

**1JT17IS015**

ABSTRACT

The project has been implemented on eclipse platform ,with the operating system windows. **Indexing** is about searching of an element in a given data set of any variable length with the help of primary key or secondary key which are some of the indexing values and to retrieve the corresponding attribute values.

The main purpose of this project is to search a required data set in a large file and gathering the information corresponding to the key value.

Depending on the search which and primary index or secondary index , a uniquely generated or a common data set can be retrieved. So the required software and hardware are easy to work.

TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| SL NO. | DESCRIPTION | PAGE NO. |
| 1. | INTRODUCTION |  |
| 1.1 | INTRODUCTION TO FILE STRUCTURE | 1 |
| 1.2 | INTRODUCTION TO INDEXING | 2 |
| 1.3 | ABOUT THE FILE | 3 |
| 1.4 | INTRODUCTION TO BANK MANAGEMENT SYSTEM | 4 |
| 1.5 | SCOPE AND IMPORTANCE OF WORK | 4 |
| 1.6 | TOOLS USED | 5 |
| 2 | IMPLEMENTATION |  |
| 2.1 | ALGORITHM | 6-7 |
| 2.2 | PROBLEM FACED DURING IMPLEMENTATION |  |
| 3 | INDEXING ANALYSIS |  |
| 3.1 | ANALYSIS | 8 |
| 4 | RESULTS AND SNAPSHOTS |  |
| 4.1 | Snapshots | 9-13 |
| 4.2 | result | 14 |
| 5 | CONCLUSIONS AND REFERENCES |  |
| 5.1 | Conclusions | 15 |
| 5.2 | References | 15 |

**CHAPTER 1**

**INTRODUCTION**

* 1. **INTRODUCTION TO FILE STRUCTURE**

File Structures is the Organization of Data in Secondary Storage Device in such a way that minimizes the access time and the storage space. A File Structure is a combination of representations for data in files and of operations for accessing the data. File Structure allows applications to read, write and modify data. It also supports finding the data that matches some search criteria or reading through the data in some particular order.

The goal of File Structure is to get the information we need with one access to the disk. If it is not possible, then get the information with as few accesses as possible. Group information so that we are likely to get everything we need with only one trip of the disk. It is relatively easy to come up with File Structure designs that meet the general goals when the files never change. When files grow or shrink when information is added and deleted, it is much more difficult.

Goal of this course is with reference to time and space is to first minimize number of trips to the disk in order to get desired information. Ideally get what we need in one disk access or get it with as few diskaccesses as possible. Secondly grouping related information so that we are likely to get everything we need with only one trip to the disk for example name, address, phone number, account balance.

Good File Structure design must have:

* Fast access to great capacity.
* Reduce the number of disk accesses.
* By collecting data into buffers, blocks or buckets.
* Manage growth by splitting these collections.

**1**

* 1. **Introduction to indexing**

File Structure design:

* In the beginning the file access was sequential, and the cost of access grew in direct proportional to the size of the file. So Indexes were added to files.
* Simple indexes use simple arrays which is sorted in main memory.
* An index lets us impose order on a file without rearranging the file.
* Indexes provides multiple access paths to a file- multiple indexes (like a particular catalog providing search for its attributes).
* An index can provide keyed access to variable-length record files.
* There are various operations in order to maintain an indexed file such as creating the data files, loading the Index file to memory before using it ,rewrite the index file , add records to data file , delete records , update the index.
* Key field is the part of index that contains keys.
* Reference field is a part of the index that contains information to locaterecords. Primary keys are unique and are used to identify the records.Secondary key is the key other than the primary key and the index built on secondary key is secondary index.

**2**

* 1. **About the file**

A File is an object on a computer that stores data, information, settings, or commands used with a computer program. In a graphical user interface such as Microsoft Windows, files display as icons that relate to the program that opens the file. For example, the picture is an icon associated with adobe acrobat PDF files, if the file was on your computer, double clicking the icon in Windows would open that file in adobe acrobat or the PDF reader installed on the computer.

A File is created using a software program on the computer. For example, to create a text file we use text editor, to create an image file we use an image editor, and to create a document we use a word processor.

Files are not made for just reading thecontents, we can also perform some operations on the Files.

* **Read operation**: Meant to read the information which is stored into the files.
* **Write operation**: For inserting some new contents into a file.
* Rename or Change the Name of the file.
* Copy the file from one location to the other.
* Sorting or arrange the contents of the file.
* Move or cut the file from one place to another.
* Delete a file.
* Execute and display the output.

We can also link a file with any other File. These are also called Symbolic Links, in the symbolic links all the files are linked by using some text alias.

**3**

* 1. **Introduction to Bank Management System**

During the past several decades personnel function has been transformed from a relatively obscure recordkeeping staff to central and top level management function. There are many factors that have influenced this transformation like technological advances, professionalism and general recognition of human beings as most important resources.

A computer based management system is designed to handle the primary information using data files and to operate on it.

This project intends to introduce a various activities such as record creation, updation ,maintainence, searching and deletion.

Here creation, searching, deletion are done using the types indexes of file structures such as

* simple index
* Secondary index
  1. **Scope and Importance of Work**

The scope of the project is to access the record of account holders in

minimum access to the disk. This project retrieves data, search, insert, delete

and modify the records by taking less time thus reducing the access to the

disk.

In this application we are able to save the records related to bank and the

Account holders who have their account.

This application is important. As the earlier method of accessing and saving

the data records would take a lot of time and more access to the disk. With the

help of indexing, the indexes are extracted and stored in main memory. So

only the specific indexed records are retrieved.

**4**

**1.5.1 Existing application and need for this application**

The existing application takes multiple accesses to the disk to retrieve and to

perform other operations on the bank dataset. This takes a lot of time.

But this application minimizes the access to the disk. Time taken is less.

The advantages of this application are:

Easy to retrieve the data

Easy to search

Modification is simpler

Minimum accesses to the disk

**1.6.Tools used**

Software :

OS : windows 10

Java eclipse

Hardware :

RAM: 2.00GB and more

**5**

**CHAPTER 2**

**IMPLEMENTATION**

**CHAPTER 2 IMPLEMENTATION**

**2.1 Algorithm**

Step 1: We need to create a file containing a particular dataset

Step 2: We need to take a search key as input

Step 3: Output: Efficiently returns a collection of matching records.

Step 4: The first column is the search key that contains the copy of the key

index.

Step 5: The second column is the pointer which contains a set of pointers

holding the address of the disk block where that particular key value is found.

Step 6: Record addition: This consists of appending the data file and inserting

a new record. The rearrangement of the index consists of sliding down the

records with keys larger than thee inserted key and then placing new records

in the opened space.

Step 7: Record Deletion: This should use the techniques of reclaiming the

space in file when deleting from the file. We must delete the corresponding

entry from the index. Shift all records with key larger than key of the deleted

record to the previous position in memory or make the index entry as deleted

by using special character as \*,- ,etc.

Step 8: In my record file the primary index is book id and the secondary

indexes are book name and author name.

Step 9: Record addition in secondary indexing: When adding a record entry

must also be added to the secondary key indexes. There may be duplicates in

secondary key, keep duplicates in sorted order of primary key.



Fig 2.1:Parts of indexes. Search key and Reference part

Step 10: Record deletion in secondary indexing: Deleting a record implies

removing all the references to the record in primary index and in all secondary

indexes. When accessing the file through secondary key, the primary indexed

file will be checked and a deleted record can be identified. 6

Step 11:Record Modification :Modifying for the data records is done by

searching based on primary index. After that choice is given to modify the

desired field. If the length of the new field is less than old field, the data is

written in that position. If it exceeds then, the older data record is deleted or

marked deleted and write the new record at the end of the file.

Step 12:Search :It allows binary search to obtain a keyed access to a record in

variable length record file.

Step 13:Time taken for dataset has been calculated for each functionality.

**2.2 Problems faced during the implementation**

There were many problems faced by me during this implementation :

Firstly, the csv file was not imported correctly and after imported the

data was not retrieved properly.

The next problem I faced was searching from 1 lakh data. There was an

exception displayed while searching. There were many attempts that

went in vain. Finally with the usage of trim() , this exception was

solved and the search function worked.

There were many small problems that were raised during

implementation and those were solved at the moment itself.

There was an exception displayed while deleting. That was raised due

to space in the string and that string was converted to long which is not

valid input. This exception was also solved using trim() function.

7

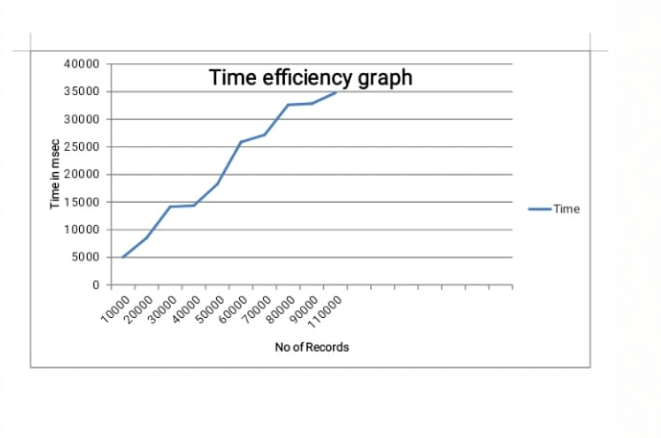
**CHAPTER 3**

**INDEXING ANALYSIS**

**CHAPTER 3 INDEXING ANALYSIS**

**3.1 Analysis**

Time efficiency for this application is showed by the below graph:



From the above graph, it is observed that as we increase the number of records,

the time in milli seconds were also increased. So by this we can say that the

time required for indexing is directly proportional to the number of input

records. Here are some cases that acts as an example for this explanation:

For 10,000 records, the time taken to form the index is 4860ms.

As we increase the number to 20,000 records, the time was also

increased to 8423ms.

When number of records were 30,000, the time for indexing was

13945ms

When number of records were 40,000, the time for indexing was 14262

ms.

When number of records were 50,000, the time for indexing was

18175ms

When number of records were 60,000, the time taken for indexing was

25770ms.

When the number of records were 70,000, the time taken for indexing

was 27130ms.

When the number of records were 80,000, the time taken was 32515ms.

When the number of records were 90,000, the time taken was 32677ms.

When the number of records were approximately 1 lakh, the time taken

was 34598ms.

So analysis that can be drawn from the above observation is that number of

records and time required for indexing are directly related.

This time efficiency graph can also be represented in the form of bar graph.

8

**CHAPTER-4**

**RESULTS AND SNAPSHOTS**

**CHAPTER4 RESULTS AND SNAPSHOTS**

**4.1.** Snapshots

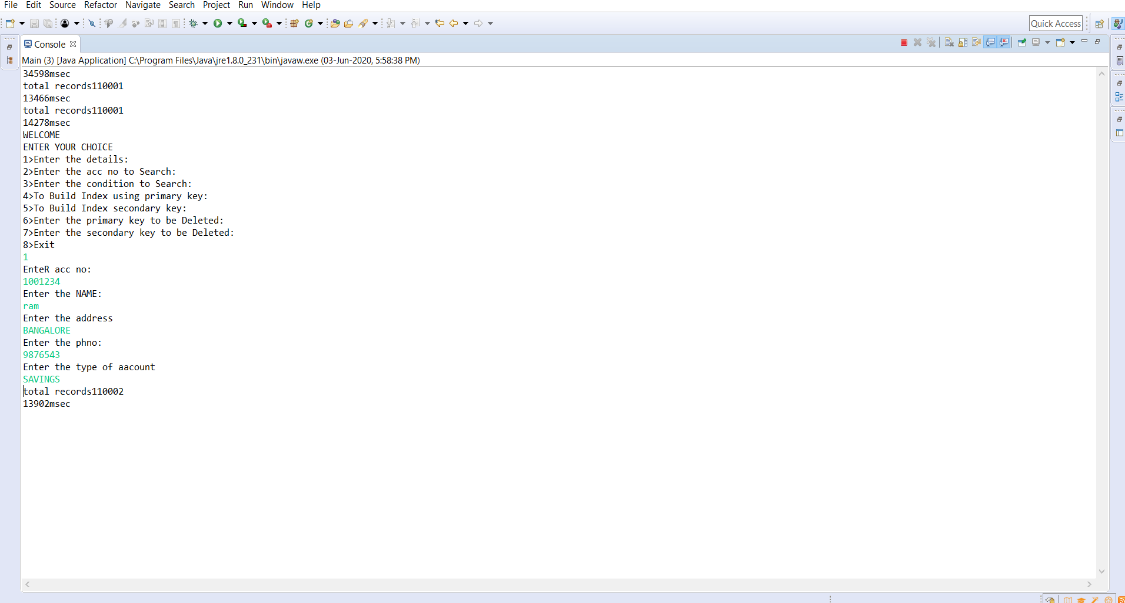


Fig 4.11. insertion of record to the file.

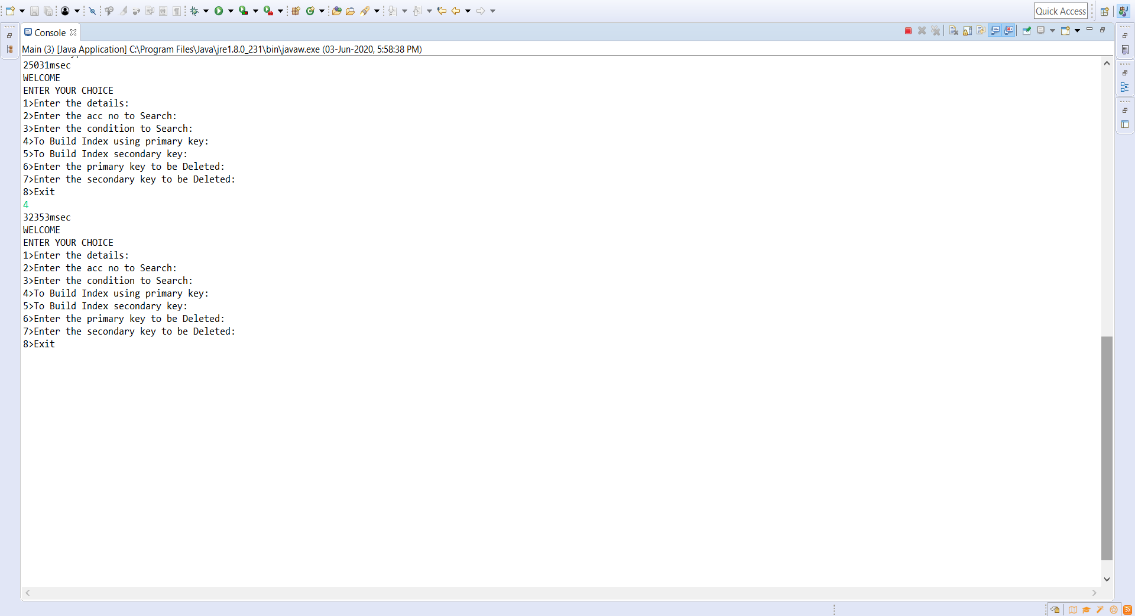


Fig 4.12. Updation of primary index.

9

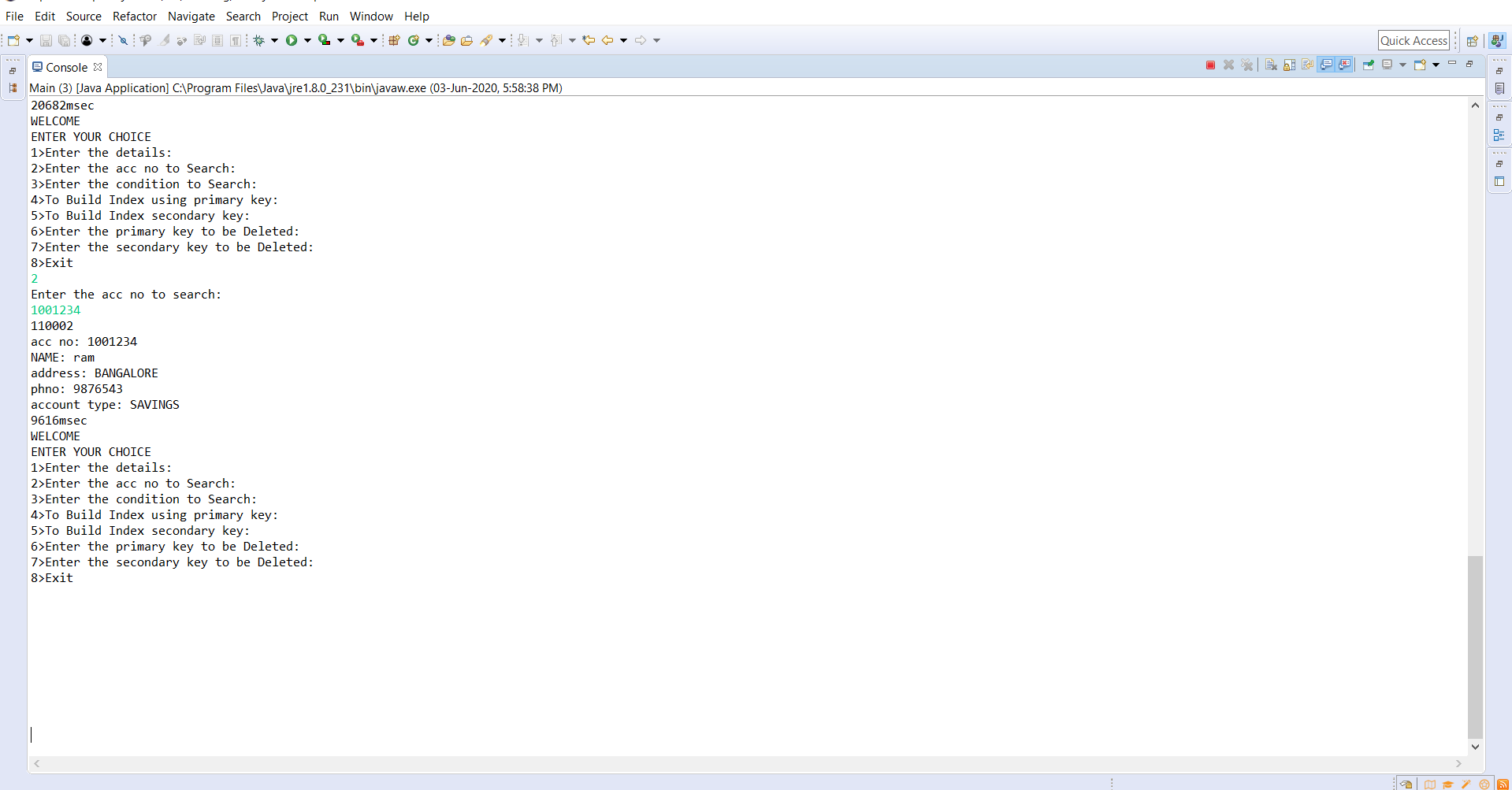


Fig 4.13. searching of account number: this is a simple index search where ,for a given account number, if the record is present, then it is displayed.

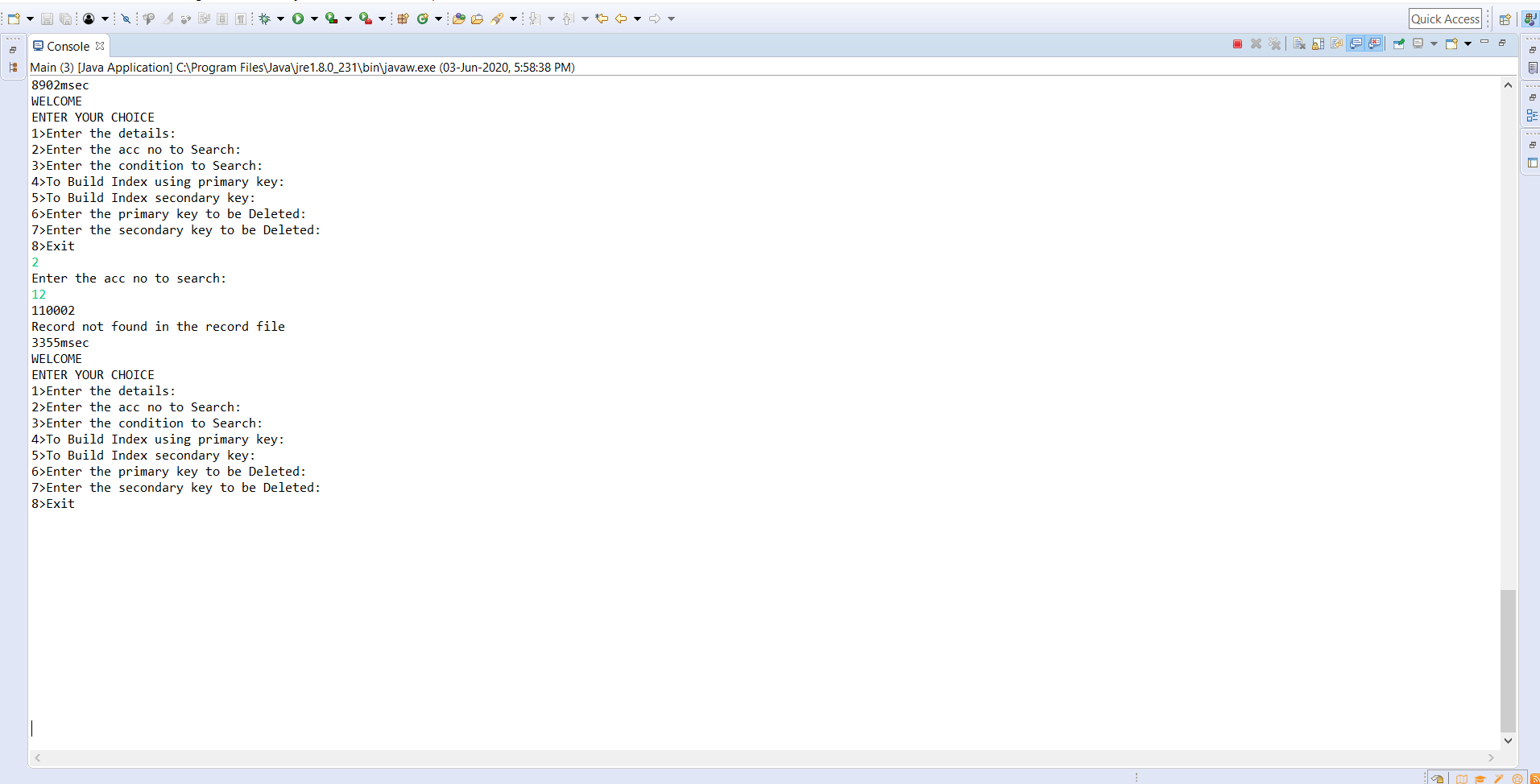


Fig 4.14. searching of account number: this is a simple index search where ,for a given account number, the record is not present, so record not found is displayed.

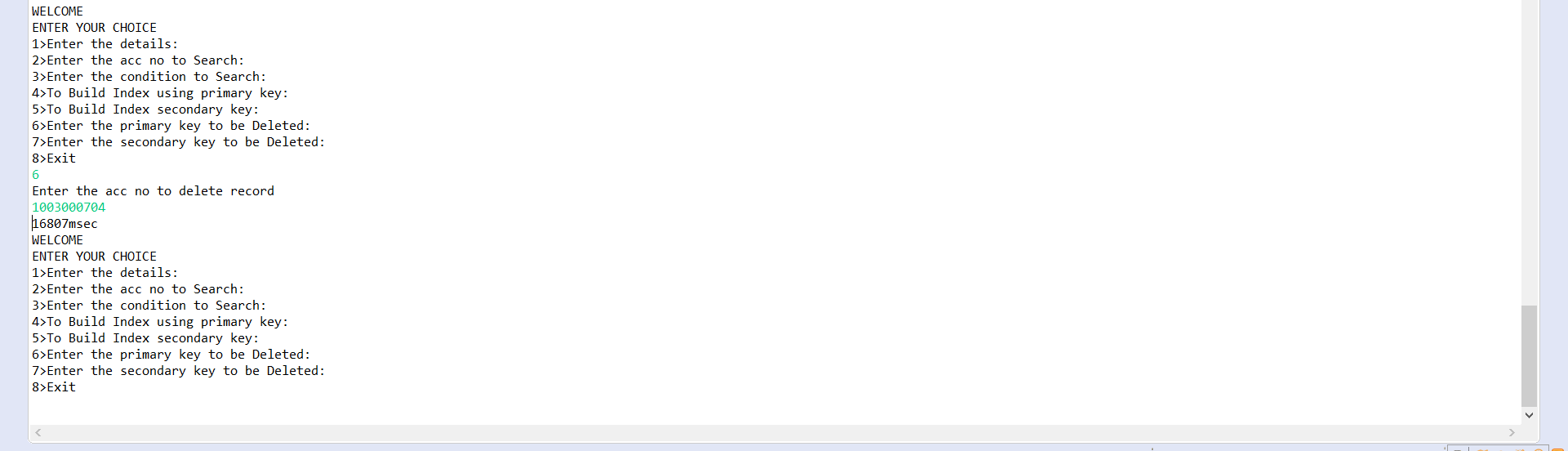


Fig 4.15. using primary key, the existing account number is deleted. 10

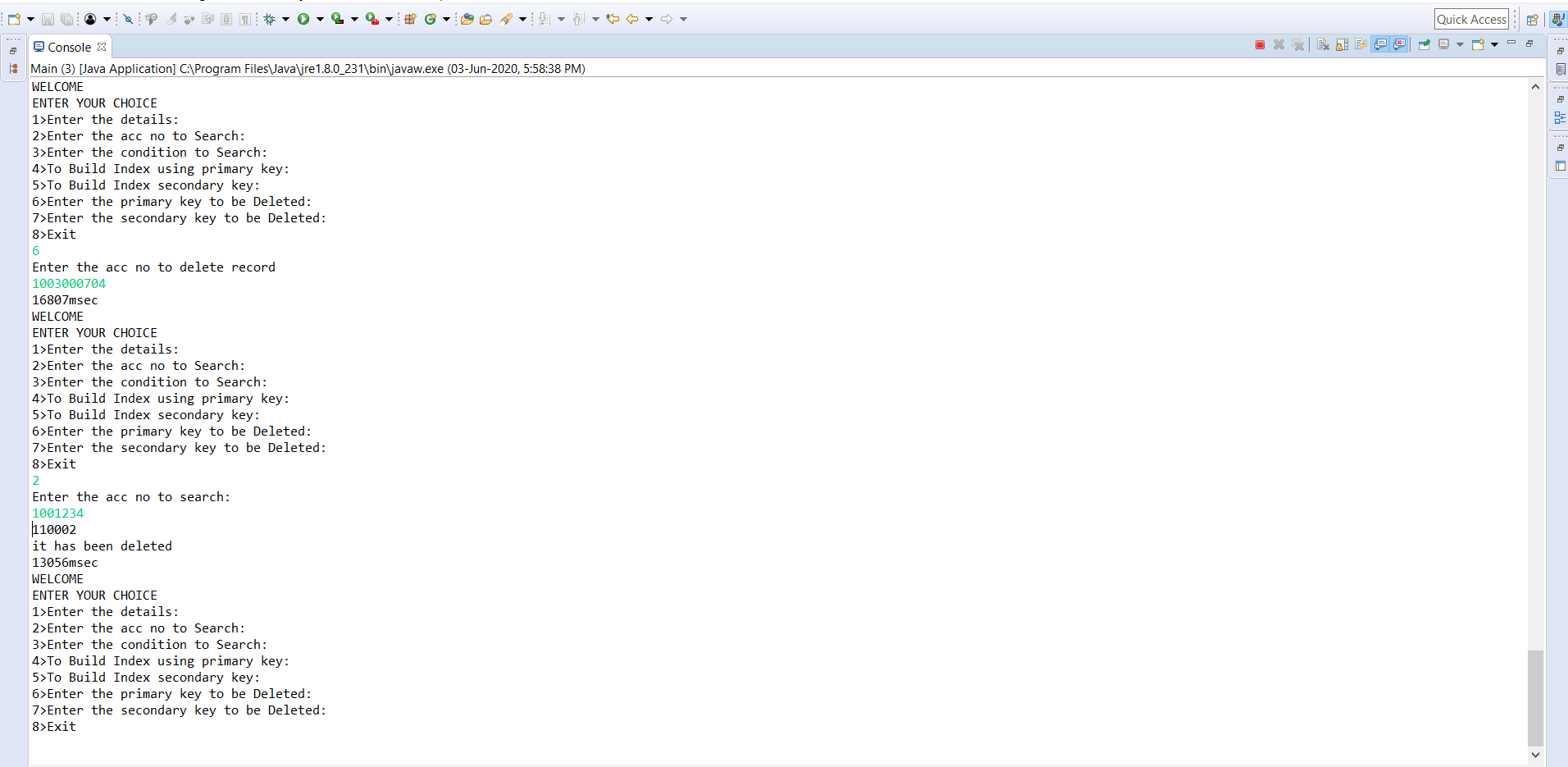


Fig 4.16. record not found in the file: here if the recorded is deleted using primary index and if it is tried to display then message this message is displayed.

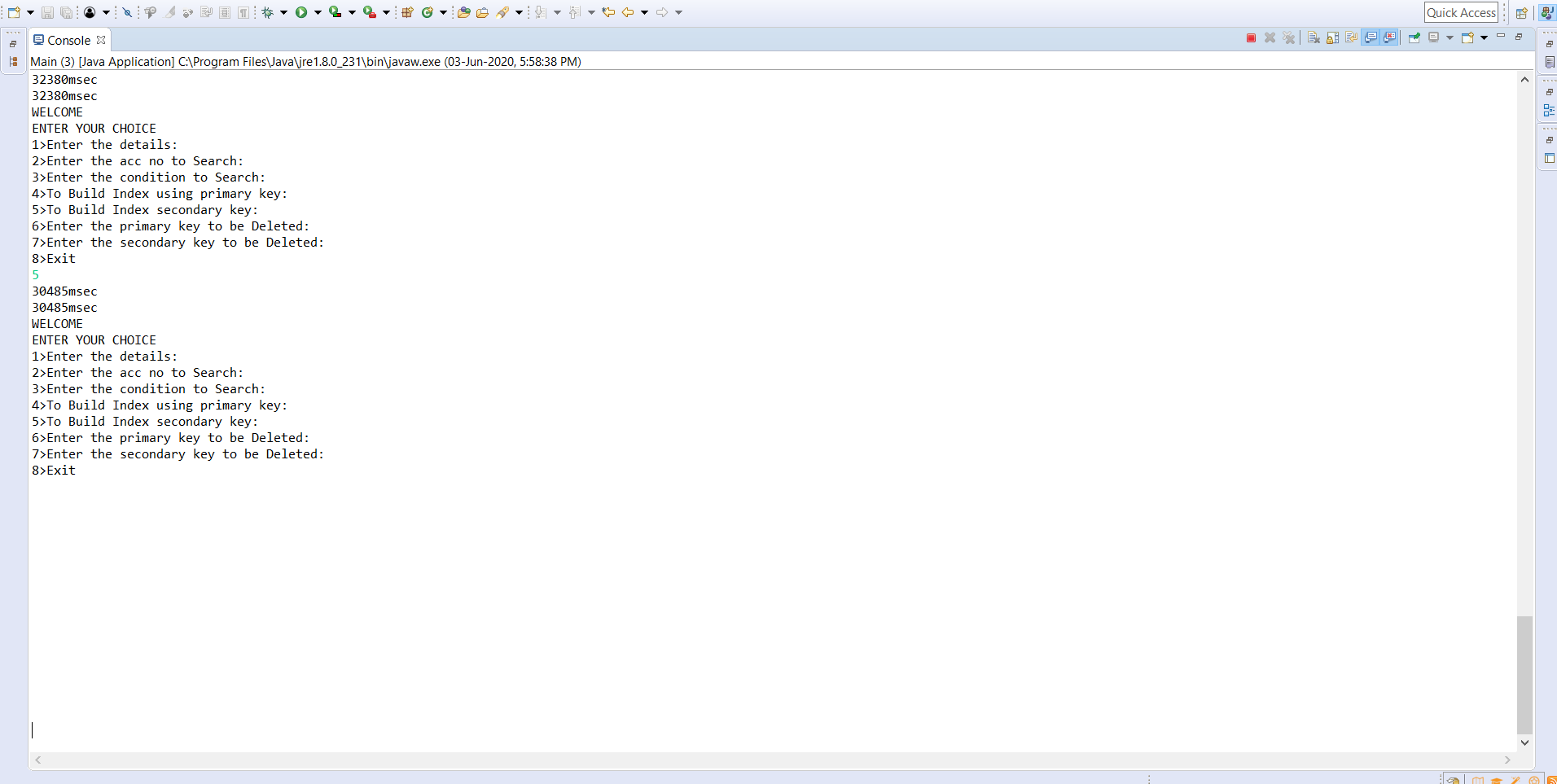


Fig 4.17. Updation of secondary index.

11

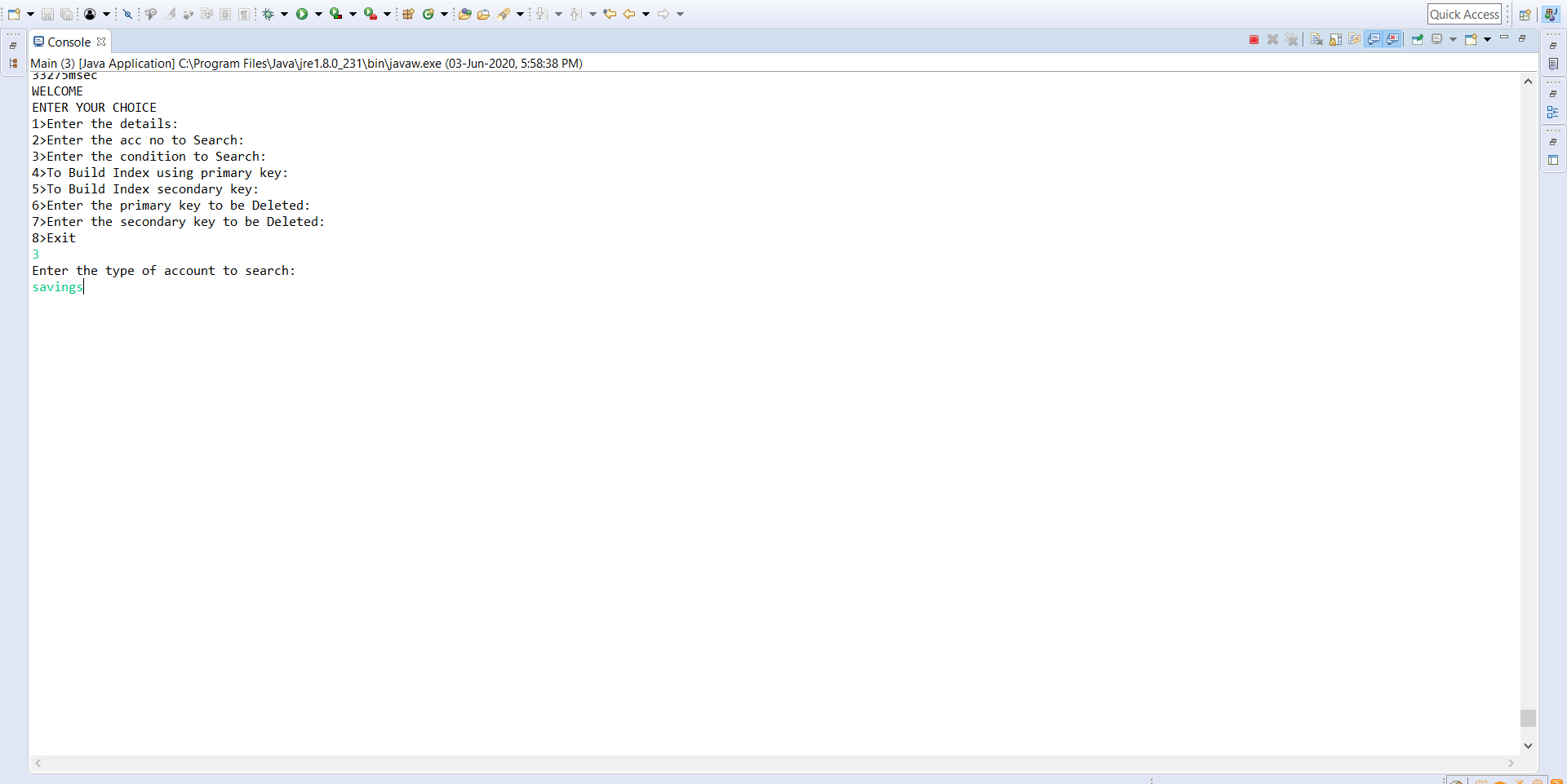


Fig 4.18. searching of data record using secondary index.

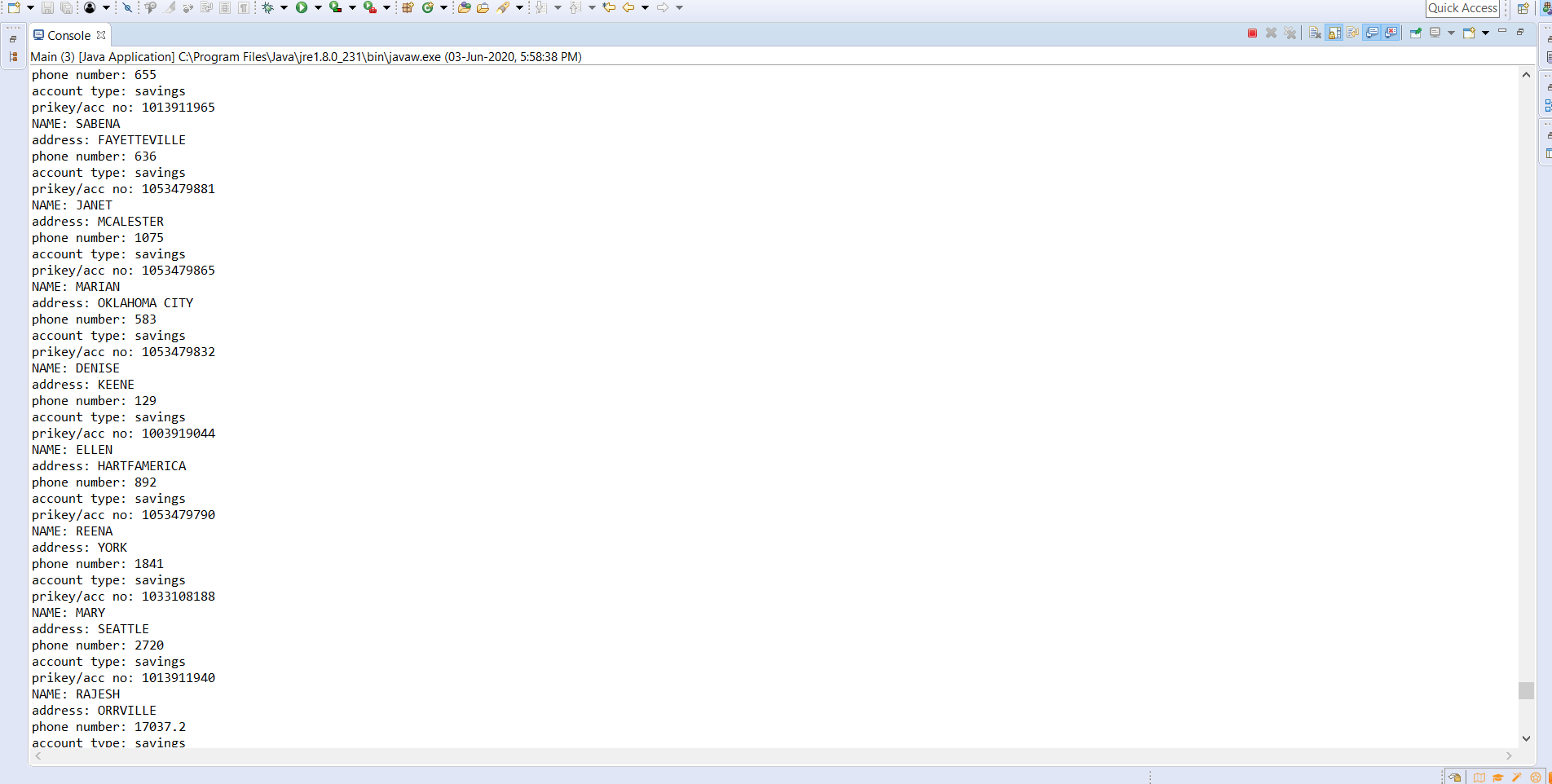


Fig 4.19. based on the account type , the data records will be displayed which are present in the file. Where the account type may be current account or savings account.

12

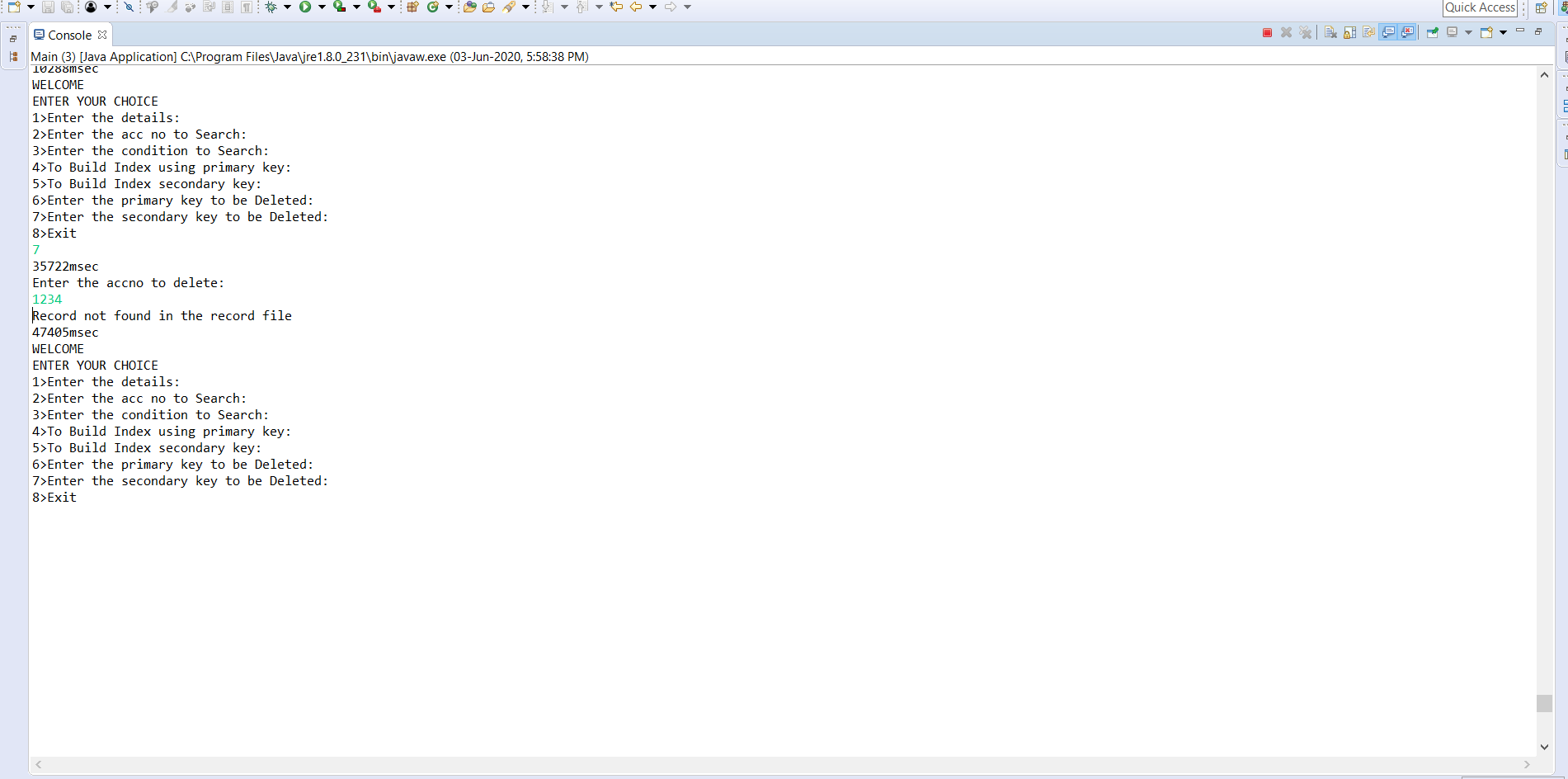


Fig 4.20. record has been deleted by secondary index hence it is not found in the file.

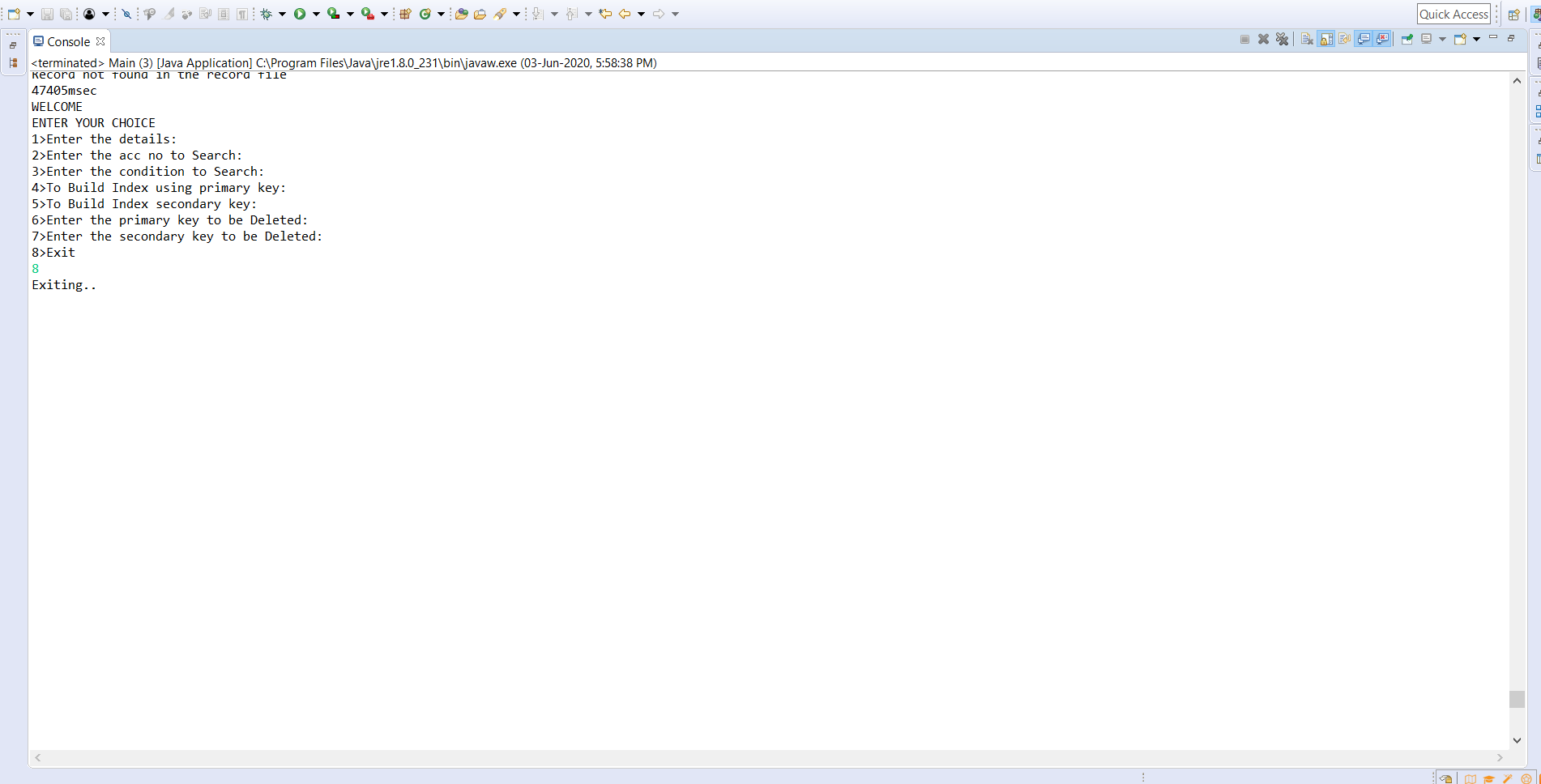


Fig 4.21. when option 8 is selected then the file is closed .

13

**4.2 Result**

This project solves the purpose for which it was designed. The operations that

can be done using this project are:

Insertion of new record to the existing file.

Searching using both primary index(accno) and secondary

index(acc\_type)

Deletion of the existing records using primary index and secondary

indexes.

These can be achieved with few accesses to the disk and minimum time.

There are few observations in the form of snapshots attached.

14

**CHAPTER-5**

**CONCLUSION & REFERENCES**

**CHAPTER 5 CONCLUSION AND REFERENCES**

**5.1 Conclusion**

We can maintain the record details of bank. There is a scope for introducing a

methods to insert a account which in turn can be developed as bank management system.

The purpose of the project was to build an application to reduce the manual work for managing bank details. This application is used to store bank details where in it holds some operations such as insertion, deletion, selection and modifications and so on. This application takes minimum access, and any future enhancements can be done.

**5.2 References**

1. JAVA TUTORIAL-TUTORIALSPOINT :This site has helped me in

learning some programming techniques in java. This helped me to

implement few operations to the project like insertion, deletion, selection

and modification.

2. JAVAPOINT: This site has helped me to learn of how to import csv file to

eclipse.

3. GEEKSFORGEEKS: This site has helped me to understand and implement

few theoretical concepts related to indexing to the project.

4. youtube videos: it helped me to clarify my doubts and many more.

5.File structures: reference book

15