**PROJECT BASED LAB REPORT**

On

**FACILITIES AT KLU INFORMATION SYSTEM**

**Submitted in partial fulfilment of the**

**Requirements for the award of the Degree of**

**Bachelor of Technology**

In

**Electronics & communication Engineering**

BY

**ROHINI PANDIRI 2100031934**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**K L University**

Green Fields, Vaddeswaram, Guntur district-522 502

**2021-2022.**

**K L University**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**



CERTIFICATE

This is to certify that the course based project titled **“FACILITIES AT KLU INFORMATION SYSTEM”**, being submitted by ROHINI PANDIRI 2100031934 inpartial fulfillment for the award of degree in **Bachelor of Technology** in **Electronics and communication Engineering** during the academic year **2021-2022.**

**Faculty In Charge Head of the Department**

**DR. BHUPESH Dr. A.C.S. SASTRY**

**ACKNOWLEDGEMENTS**

My sincere thanks to **DR. D. BHUPESH** in the Lab for their outstanding support throughout the project for the successful completion of the work

We express our gratitude to **Dr. A.C.S. SASTRY,** Head of the Department for Electronics and communication Engineering for providing us with adequate facilities, ways and means by which we are able to complete this project based lab.

We would like to place on record the deep sense of gratitude to the honourable Vice Chancellor, K L University for providing the necessary facilities to carry the concluded term paper work.

we thank all Teaching and Non-Teaching Staff of our department and especially my classmates and my friends for their support in the completion of our term paper work.

**ROHINI PANDIRI**

**2100031934**

**CONTENTS**

**s.no content**

**1. ABSTRACT**

**2. INTRODUCTION**

**3. FUNCTIONAL REQUIREMENTS**

**4. NON-FUNCTIONAL REQUIREMENTS**

**5. CODE**

**6.          RESULT**

**ABSTRACT**

In college information system, we can easily get the information about the admission, seats and vacancies in a particular course and college through a single click and also fee can be paid easily. This project is based on carrying out various tasks which goes under college management. It will control all activities for a particular college. To handle all the tasks, system has been divided into different modules and presented on a single window, so that its user can handle it in eco-friendly manner. It display’s all the information about the college. All the modules in college administration are interdependent. They are maintained manually. So they need to be automated and centralized as, Information from one module will be needed by other modules. In this project we are going to explain about these modules

* Library
* Transport system
* Hostel
* Stationary
* Canteen
* Facility

**INTRODUCTION:**

1.1 What is college Information system ?

The college Information system is meant for maintaining information of various facilities like hostels, datacentre, library, physical education, transport etc.. It facilitates to display this information for the benefit and use of students and parents.

1.2 Problem Definition

The manual distribution and display of library college facilities information is very difficult. In a day there may be hundreds or thousands of persons used to ask about the facilities of the college and the hostel. The time required for this will be more and the parents must travel to college to get this information from the authorities.

The klu Facilities Information System will display the number of college facilities like maintenance of library, hostels, data centre, transport, physical education, soft skills, laboratories etc. By this we can maintain a record of the facilities and also reduce the time for the parents to know more about college.

1.3 Platform Requirements

|  |  |  |
| --- | --- | --- |
| Hardware/Software | Hardware / Software element | Specification /version |
| Hardware | Processor | Intel core to duo |
| RAM | 1 GB |
| Hard Disk | 100 GB |
| Software | OS | Windows XP |
| Java and Netbeans IDE |  |

**DESCRIPTION:**

The project is done based on java programing. This is through object oriented programing. This gives some necessary information about kl university facilities. When we want to search any information regarding  kl university facilities.  It is very difficult to find the facilities information at kl university . we can get all the information of any facility at kl university immediately and accurately. Here almost all work is computerized. So the accuracy is maintained. Maintaining backup is very easy .some of the issues can be solved easily by using this project.

The project  kl university facilities information system deals with the information of every state which includes Facilities  , Transport , Canteen ,Library . It stores all the data and details of  kl university facilitieis difficult using the manual system as the information as necessary information cannot be available. It can be used in society in various systems.

**FUNCTIONAL REQUIREMENTS**

**Purpose**  
The purpose of doing this  project is  to provide all the requirements for the KL UNIVERISITY  information. System. The user can search easily anything regarding the hospitals he want with moving here and there .This will make his time manageable and there will no waste of time in goingout.  

|  |  |  |  |
| --- | --- | --- | --- |
| 1 |  | KL UNIVERISITY  information |  |
|  |  | To search for kl university facilities information system the user need | Here we will be creating the list of details of kl university facilities showing the transport,canteen,library,classes,stationery etc. |
|  |  | Displays all facility  details that kl university provided | This  is used maintain the details of the kl university facilities and college details for the use |
|  |  | To search facilities in the kl university | This is used to search a particular facilities details when user gives a particular search number of that facility name. |
|  |  | To generate  details | Here in generating reports we will be able to display all the details of records of  kl university facility with details in the form of files. |
| 2 |  | Main Module |  |
|  |  | To display System | After all modules were integrated through files we display the complete information. |
|  |  | To process menu | Here **Processing Menu** can be done through if else statements  with all the modules that are present in the file. |
|  |  | Initializations | Here **Initializations** can be done to methods, variables  as well as objects which are present in classes |
|  |  | To access | To provide easy access we informed every thing to user which  will able to access his details . we can be able to read the file and record of that file, and write that file where he can be able to do modifications for files that are present .. |

**NON-FUNCTIONAL REQUIREMENTS**

INTERFACE:

An interface is a reference type in Java. It is similar to class. It is a collection of abstract

methods. A class implements an interface, thereby inheriting the abstract methods of

the interface. Along with abstract methods, an interface may also contain constants, default

methods, static methods, and nested types.

INHERITANCE :

Inheritance in java is a mechanism in which one object acquires all the properties and

behaviors of parent object. The idea behind inheritance in java is that you can create new

classes that are built upon existing classes.

ABSTRACT CLASS:

A class that is declared with abstract keyword, is known as abstract class in java. It can have abstract and non-abstract methods (method with body).**Abstraction** is a process of hiding the implementation details and showing only functionality to the user. Another way, it shows only important things to the user and hides the internal details for example sending sms, you just type the text and send the message. You don't know the internal processing about the message delivery.

ABSTRACT METHOD:

An abstract method is a method that is declared, but contains no implementation. Abstract classes may not be instantiated, and require subclasses to provide implementations for the abstract methods. Let's look at an example of an **abstract** class, and an abstract method.

STATIC KEYWORD:

All instances share the same copy of the variable. A class variable can be accessed directly with the class, without the need to create a instance.

EXCEPTION HANDLING:

An exception (or exceptional event) is a problem that arises during the execution of a program. When an Exception occurs the normal flow of the program is disrupted and the program/Application terminates abnormally, which is not recommended, therefore, these exceptions are to be handled.

**SOURCE CODE OF THE PROJECT**

**import java.io.\*;**

**import java.util.\*;**

**class Klu**

**{**

**void bus()**

**{**

**try**

**{**

**FileInputStream fin=new FileInputStream("bus.java");**

**int i=0;**

**while((i=fin.read())!=-1)**

**System.out.print((char)i);**

**fin.close();**

**}**

**catch(Exception e)**

**{**

**System.out.println(e);**

**}**

**}**

**void library()**

**{**

**try**

**{**

**FileInputStream fin=new FileInputStream("library.java");**

**int i=0;**

**while((i=fin.read())!=-1)**

**System.out.print((char)i);**

**fin.close();**

**}**

**catch(Exception e)**

**{**

**System.out.println(e);**

**}**

**}**

**void canteen()**

**{**

**try**

**{**

**FileInputStream fin=new FileInputStream("canteen.java");**

**int i=0;**

**while((i=fin.read())!=-1)**

**System.out.print((char)i);**

**fin.close();**

**}**

**catch(Exception e)**

**{**

**System.out.println(e);**

**}**

**}**

**void stationary()**

**{**

**try**

**{**

**FileInputStream fin=new FileInputStream("stationary.java");**

**int i=0;**

**while((i=fin.read())!=-1)**

**System.out.print((char)i);**

**fin.close();**

**}**

**catch(Exception e)**

**{**

**System.out.println(e);**

**}**

**}**

**void courses()**

**{**

**try**

**{**

**FileInputStream fin=new FileInputStream("courses.java");**

**int i=0;**

**while((i=fin.read())!=-1)**

**System.out.print((char)i);**

**fin.close();**

**}**

**catch(Exception e)**

**{**

**System.out.println(e);**

**}**

**}**

**void placements()**

**{**

**try**

**{**

**FileInputStream fin=new FileInputStream("library.java");**

**int i=0;**

**while((i=fin.read())!=-1)**

**System.out.print((char)i);**

**fin.close();**

**}**

**catch(Exception e)**

**{**

**System.out.println(e);**

**}**

**}**

**void classes()**

**{**

**try**

**{**

**FileInputStream fin=new FileInputStream("stationary.java");**

**int i=0;**

**while((i=fin.read())!=-1)**

**System.out.print((char)i);**

**fin.close();**

**}**

**catch(Exception e)**

**{**

**System.out.println(e);**

**}**

**}**

**void display()**

**{**

**try**

**{**

**FileInputStream fin=new FileInputStream("disp.java");**

**int i=0;**

**while((i=fin.read())!=-1)**

**System.out.print((char)i);**

**fin.close();**

**}**

**catch(Exception e)**

**{**

**System.out.println(e);**

**}**

**}**

**}**

**class Kluni**

**{**

**public static void main(String args[])**

**{**

**Scanner s=new Scanner(System.in);**

**Klu O=new Klu();**

**O.display();**

**int n;**

**do**

**{**

**System.out.print("\n\n\t\t enter your choice:");**

**n=s.nextInt();**

**switch(n)**

**{**

**case 1:O.bus();**

**break;**

**case 2:O.library();**

**break;**

**case 3:O.canteen();**

**break;**

**case 4: O.courses();**

**break;**

**case 5:O.stationary();**

**break;**

**case 6:O.placements();**

**break;**

**case 7:break;**

**}**

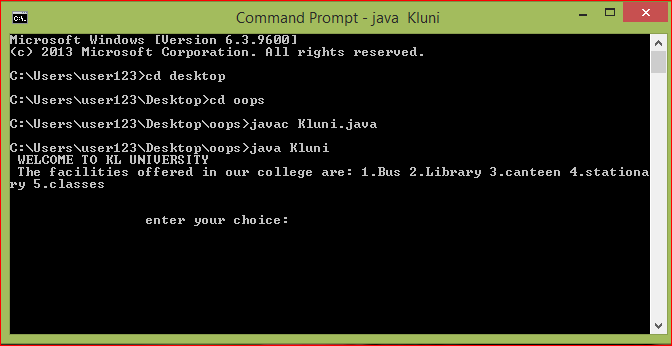
**}while(n!=7);**

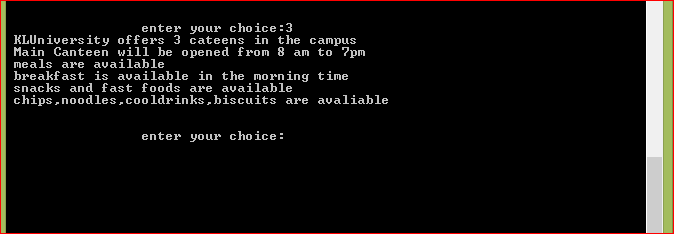
**}**

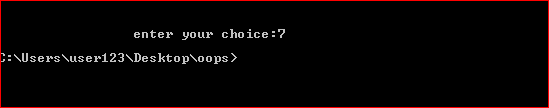
**}**

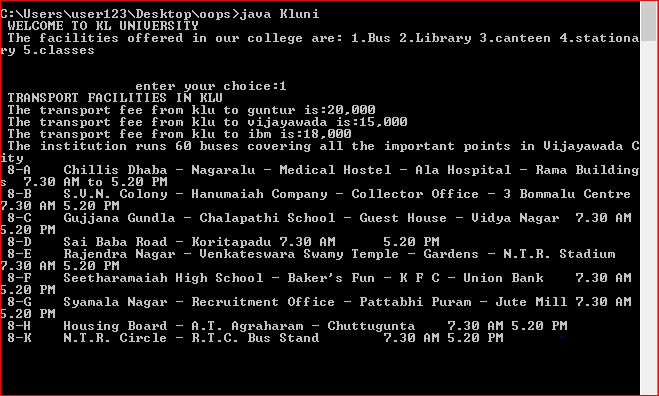
**RESULTS**

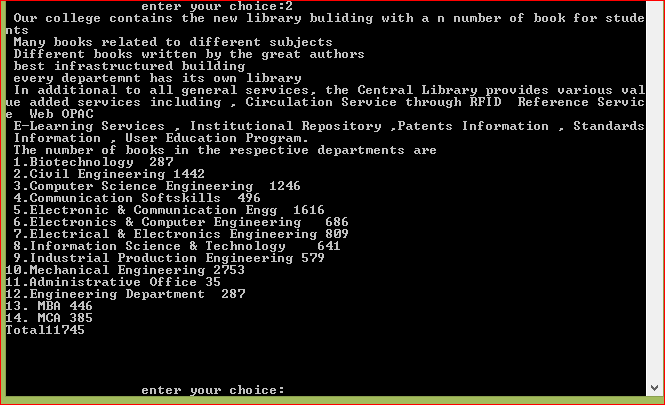
All programs are compiled successfully. All programs are run and tested for correctness. Results obtained are:

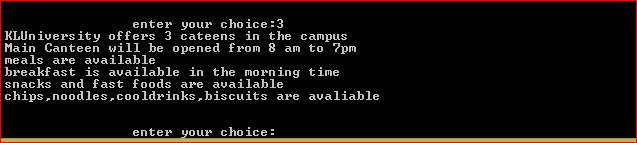


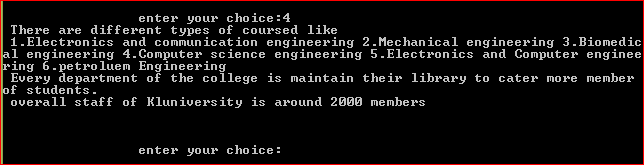


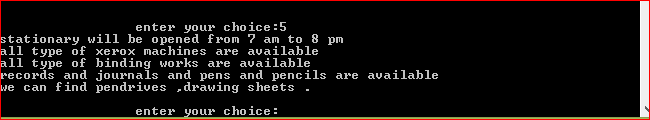


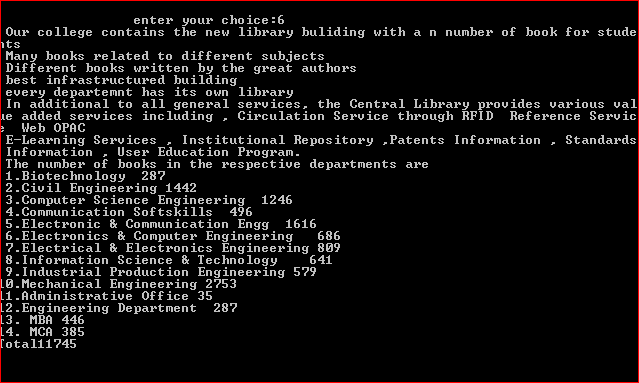












**4 . Conclusion and Future Scope**

4.1 Conclusion

The project is successfully completed to the extent possible. The results of the project are shown earlier.

4.2 Future Scope: Some more functions or modules may be added to project. After studying and understanding Graphic User Interface (GUI) of java, the inputs and outputs can be improved and implemented using GUI.

5. **Bibliography**

* Herbert Schildt, “The Complete Reference Java2”, 5th edition TMH, 2002.
* Timothy A. Budd, “An Introduction to Object-Oriented Programming”, 3/E, Pearson, 2008.
* Jim Keogh, “The Complete Reference J2EE”, TMH, 2006.
* Deitel & Deitel, “‘JAVA – How to program”, 6th edition, PHI, 2007.