#### HBTU-logo

#### ECS-202 : SOFTWARE ENGINEERING

**PROJECT:** Software Component Cataloguing Software

#### Contributors:

#### Submitted to- Aryan Verma : 17CS13

#### Dr. Mousumi Dhara Ashish Singh : 17CS14

#### Ayush Kumar : 17CS16

#### Ayushman Yadav : 17CS17

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and our college, ***Harcourt Butler Technical University.***We would like to extend our sincere thanks to all of them.

We are highly indebted to **Dr. Mousumi Dhara**  for her guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our gratitude towards our parents for their kind co-operation and encouragement which helped us in completion of this project.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

***Software Component Cataloguing Software***

**Introduction**

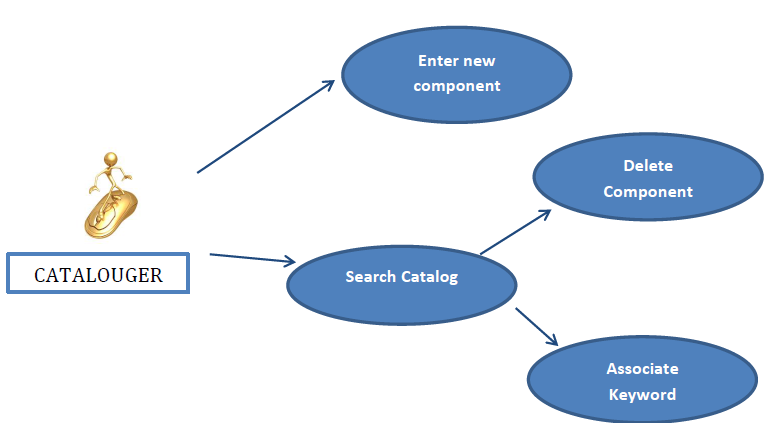
#### The Software Component Cataloguing Software

#### Consists of a software components catalogue and various functions defined on this components catalogue. The software components catalogue holds details of the components which are potentially reusable. A cataloguer may enter components in the catalogue, may delete components from the catalogue, and may associate reuse information with a catalogue component in the form of a set of key words.

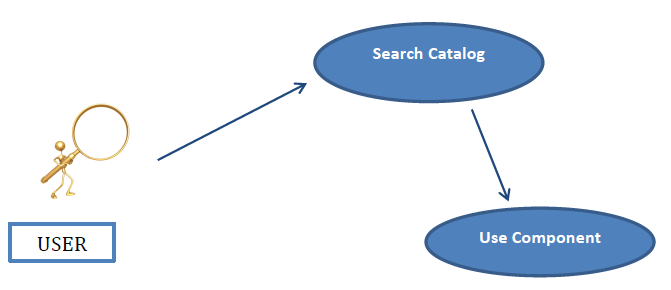
A user of the catalogue may query about the availability of a component using certain key words to describe the component. In order to help manage the component catalogue (i.e., periodically purge the unused components) the cataloguing software maintains information such as how many times a component has been used, and how many times the component has come up in a query but not used. Components are classified into categories hierarchically. A user can browse the components in each category.

**Objective**

The reusable components can be either design or code. The design might have been constructed using different design notations such as UML, ERD, structured design, etc. Similarly, the code might have been written using different programming languages. A cataloguer may enter components in the catalogue, may delete components from the catalogue, and may associate reuse information with a catalogue component in the form of a set of key words. A user of the catalogue may query about the availability of a component using certain key words to describe the component. In order to help manage the component catalogue (i.e., periodically purge the unused components) the cataloguing software should maintain information such as how many times a component has been used, and how many times the component has come up in a query but not used. Since the number of components usually tend to be very high, it is desirable to be able to classify the different types of components hierarchically. A user should be able to browse the components in each category.

**Dataflow Diagram**

|  |
| --- |
|  |



**System Requirements**

1. Operating System (i.e. Windows 8, 10)
2. Processor Speed (i.e. Core2Duo, 3.2 GHz or Power PC G5, 2.0 GHz)
3. Memory, a.k.a. RAM (i.e. 512 MB)
4. Graphics Card (i.e. ATI Radeon 9800 w/ 256 MB video memory)
5. Hard Disk Space (i.e. 80 GB available)

**Software** - Software is a set of instructions or programs written by programmers/developers on various languages for computer to perform some specific task Software is often divided into two categories

**1) System Software**-It is a base for application software which responsible for managing hardware In other words we can say that system software is a intermediary between user and hardware. System software is also known as Operating System Example MS-Windows, UNIX, Linux, Sun Solaris

**2) Application Software**- or simply applications are often called productivity programs or end-user programs because they enable the user to complete tasks, such as creating documents, spreadsheets, databases and publications, doing online research, send email, designing graphics, running businesses, and even playing games! Application software is specific to the task it is designed for and can be as simple as a calculator application or as complex as a word processing application.

The five types of systems software, are all designed to control and coordinate the procedures and functions of computer hardware. They actually enable functional interaction between hardware, software and the user.

Systems software carries out middleman tasks to ensure communication between other software and hardware to allow harmonious coexistence with the user.

Systems software can be categorized under the following:

**Operating system:** Harnesses communication between hardware, system programs, and other applications.

**Device driver:** Enables device communication with the OS and other programs.

**Firmware:** Enables device control and identification.

**Translator:** Translates high-level languages to low-level machine codes.

**Utility:** Ensures optimum functionality of devices and applications.

**1. Operating System (OS)**

The operating system is a type of system software kernel that sits between computer hardware and end user. It is installed first on a computer to allow devices and applications to be identified and therefore functional.

System software is the first layer of software to be loaded into memory every time a computer is powered up.

Suppose a user wants to write and print a report to an attached printer. A word processing application is required to accomplish this task. Data input is done using a keyboard or other input devices and then displayed on the monitor. The prepared data is then sent to the printer.

In order for the word processor, keyboard, and printer to accomplish this task, they must work with the OS, which controls input and output functions, memory management, and printer spooling.

Today, the user interacts with the operating system through the graphical user interface (GUI) on a monitor or touchscreen interface. The desktop in modern OSs is a graphical workspace, which contains menus, icons, and apps that are manipulated by the user through a mouse-driven cursor or the touch of a finger. The disk operating system (DOS) was a popular interface used in the 1980s.

**2. Device Drivers**

Driver software is a type of system software which brings computer devices and peripherals to life. Drivers make it possible for all connected components and external add-ons perform their intended tasks and as directed by the OS. Without drivers, the OS would not assign any duties.

Examples of devices which require drivers:

Mouse

Keyboard

Soundcard

Display card

Network card

**Printer**

Usually, the operating system ships with drivers for most devices already in the market. By default, input devices such as the mouse and keyboard will have their drivers installed. They may never require third-party installations.

If a device is newer than the operating system, the user may have to download drivers from manufacturer websites or alternative sources.

**3. Firmware**

Firmware is the operational software embedded within a flash, ROM, or EPROM memory chip for the OS to identify it. It directly manages and controls all activities of any single hardware.

Traditionally, firmware used to mean fixed software as denoted by the word firm. It was installed on non-volatile chips and could be upgraded only by swapping them with new, preprogrammed chips.

This was done to differentiate them from high-level software, which could be updated without having to swap components.

Today, firmware is stored in flash chips, which can be upgraded without swapping semiconductor chips.

**4. Programming Language Translators**

These are intermediate programs relied on by software programmers to translate high-level language source code to machine language code. The former is a collection of programming languages that are easy for humans to comprehend and code (i.e., Java, C++, Python, PHP, BASIC). The latter is a complex code only understood by the processor.

Popular translator languages are compilers, assemblers, and interpreters. They're usually designed by computer manufacturers. Translator programs may perform a complete translation of program codes or translate every other instruction at a time.

Machine code is written in a number system of base-2, written out in 0 or 1. This is the lowest level language possible. While seemingly meaningless to humans, the zeros and ones are actually sequenced intelligently by the processor to refer to every conceivable human code and word.

Besides simplifying the work of software developers, translators help in various design tasks. They;

Identify syntax errors during translation, thus allowing changes to be made to the code.

Provide diagnostic reports whenever the code rules are not followed.

Allocate data storage for the program.

List both source code and program details.

**5. Utilities**

Utilities are types of system software which sits between system and application software. These are programs intended for diagnostic and maintenance tasks for the computer. They come in handy to ensure the computer functions optimally. Their tasks vary from crucial data security to disk drive defragmentation.

Most are third-party tools but they may come bundled with the operating system. Third-party tools are available individually or bundled together such as with Hiren Boot CD, Ultimate Boot CD, and Kaspersky Rescue Disk.

**Types of Software Applications**

The following table describes different kinds of software applications that would be suitable for different tasks:

Word Processing software - Use this kind of tool to create worksheets, type letters, type papers, etc.

MS Word, WordPerfect, MS Works, AppleWorks, l.

Desktop Publishing software - Use this software to make signs, banners, greeting cards, illustrative worksheets, newsletters, etc.

Adobe PageMaker, MS Word, MS Publisher, AppleWorks, MS Works, Quark Express.

Spreadsheet software - Use this kind of tool to compute number-intensive problems such as budgeting, forecasting, etc. A spreadsheet will plot nice graphs very easily.

MS Excel, Quattro Pro, Lotus 1-2-3, MS Works, AppleWorks,

Database software - Use this software to store data such as address, membership and other text information. A database can be used to easily sort and organize records.

MS Access, Filemaker Pro, AppleWorks, MS Works, ...

**Presentation software** - Use this software to create multimedia stacks of cards/screens that can effectively present a lesson or a sales pitch. The user often clicks on buttons to advance to the next screen in a sequence.

MS PowerPoint, AppleWorks (slideshows), HyperStudio, Flash, Director, HyperCard, Digital Chisel, SuperCard, Corel Envoy.

Internet Browsers - This software allows one to surf the Web. Often they can read email and create Web pages too.

Netscape Navigator (or Netscape Communicator), MS Internet Explorer, AOL Browser.

Email programs - These programs send and receive email.

Netscape Messenger (part of Netscape Communicator), MS Outlook Express, MS Outlook, Eudora, AOL browser (has email built in).

Graphics Programs (pixel-based) - This software allows one to touch up photographs and create graphics from scratch.

Adobe Photoshop, Paint Shop Pro, AppleWorks, MS Works, MS Paint (comes free on Windows PC's), Painter.

Graphics Programs (vector-based) - This software creates graphics that are similar to illustrations or cartoon drawings.

Adobe Illustrator, Corel Draw, AppleWorks, MS Works, MS Word.