Introduction

For a large period of time the flowers were the best known for their beauty and fragrance. Flower is a very important part of a plant and is mostly used for identification of a plant. Flowers are available in a large number of colours, sizes and shapes. A flower plays a major role in producing new seeds for new growth. There are many similar looking plants which can be differentiated only on the basis of difference in the colour and shape of the flower as the other vegetative parts of the plant may differ in appearance due to climatic conditions, geographical location, and various biotic and abiotic stresses. Other than that a flower can also tell, if the plant is suffering from a particular diseases as flowers are the most delicate part of the plant and gets affected first.

Experienced taxonomist and botanists mostly do this identification of flower just by looking at it. Whereas naive person doesn’t have encyclopaedia in hand, when facing with flowers, so he/she doesn’t know what is it. Hence, they will have to take help either from guide books or web pages. Recognition is one of the major applications in computer vision. The majority of the research effort has been on the classification between different categories, such as between cars and airplanes as posed in the pascal. Many recent approaches have been successful in this area. However, we aim to use computer vision techniques to distinguish between visually similar objects. In particular, we investigate the possibility to separate flower species from each other. Flowers from different species may look very similar both in shape or color inter-class similarity and flowers from the same species may look different intra-class dissimilarity. Together, these facts result in a very interesting and challenging computer vision problem. But besides our ambition to tackle with difficult problems, there are also application-related interests in solving it. There are several books filled with information about flowers but it becomes quite difficult for human brain to remember all those details. It is quite difficult to search for a scientific name when we don’t have any information source and all we have is flower in our hand. The searching on Internet is restricted to keywords. Therefore the Internet approach is also not practical.

Great efforts have been undertaken over the last half century to build machines that can see. A great number of fundamental problems have been solved. Examples of such problems are the detection of edges in images, the computation of basic flow fields representing motion throughout a sequence of images, or the point-by-point registration of a pair of stereoscopic images. At the same time, one cannot escape the impression that the biggest problems are still far ahead of us. Central to understanding images is the problem of recognizing objects in images. Humans can recognize objects effortlessly and are rarely even aware of the changes in an object’s appearance that occur, for example, due to changes in viewing direction or a shadow being cast across the object. We also readily group instances of objects, such as cars, faces, shoes, or houses into a single object category and forget about the differences between the individual members. At the same time, we can still discriminate on a sub-categorical level. On the other hand, everyone who has ever dealt with a computer has inevitably experienced that even the smallest change in the information provided to a computer can, and often does, make all the difference in the world. Teaching a machine to recognize objects is all about teaching it which differences in the raw image information matter and which don’t.

A flower has many subparts which include petals, stamen/ pistil area and sepal. The four main parts of a flower are generally defined by their positions on the receptacle and not by their function. Many flowers lack some parts or parts may be modified into other functions and/or look like what is typically another part. Out of these only stamen/ pistil area is visible when a fully grown flower is seen vertically. Hence, mostly taxonomist identifies a flower by examining the features of these areas of flower. A state-of-art recognition system often takes the whole input image and extracts features either in a dense (using grid points) or sparse (using interest points) fashion. We note that the difficulty of our problem, the high similarity between flower species, implies a high similarity in the feature space as well, which lower the classification performance. We believe that this similarity can be reduced if we first localize the flower inside the image and only extract features from the flower and discard the background clutter. This idea is reflected as deleting background of flower by which only features of flower will be visible. Keeping the above given points in our mind, a system was proposed which identifies a flower through its digital image with white background. In this system features of petals and stamen/pistil area are compared through which a flower is recognized, hence recognizing the plant. Our system can help people interested in flora to identify the flowers in order to get further information about their species. This idea can be easily extended to a more general vision of a system which takes images and provides additional information about it, like a Wikipedia using image instead of text search. The features that the software uses for flower recognition includes shape, colour along with the stamen/pistil area which gives the accurate results.

The use of foreground-background segmentation in a visual classification system is a controversial issue in modern computer vision. On the one hand, the background contains information that is irrelevant in separating one class from others. This information may be converted into noise during the feature extraction and causes problem in the end classification. Several studies have reported improved classification/detection performance once accurate segmentation masks are given.

Traditional approach for flower recognition includes that the taxonomists consider the clearly visible features of flowers along with the climatic conditions of the area where that flower is found. These features include:

* Whether the flower is symmetrical i.e. looks similar from all sides or asymmetrical like orchid.
* The number of petals/ rays a flower has.
* Whether the petals are fused together or not.
* Colour of the flower petal and stamen/ pistil area.
* Whether the flower exists in a cluster or individually.
* The environment of the area where the flower is found.
* The month in which that flower blooms.

Digital image and its features

When using digital equipment to capture, store, modify and view photographic images, they must first be converted to a set of numbers in a process called digitization or scanning. Computers are very good at storing and manipulating numbers, so once your image has been digitized you can use your computer to archive, examine, alter, display, transmit, or print your photographs in an incredible variety of ways. Any image from a scanner or digital camera or stored in the computer is called digital image. A digital image is made up of many tiny dots known as pixels. Depending upon size and number of dots, the clarity of picture changes. If pixels are small the image will be clearer as the size of pixel increases image becomes more blur. The amount of details that an image can hold is controlled by the resolution of image. The word resolution describes the number of pixels that make up an image.

The data stored in an image is just a series of RGB numeric colour values in rows and columns. Each pixel displays one colour and has a particular RGB value. But in our software instead of using RGB colour model we use HSV colour model as it approximately perceives and interprets colour values just the way humans do. HSV is a color model that describes [colors](http://desktoppub.about.com/od/glossary/g/Color.htm) (hue or tint) in terms of their shade (saturation or amount of gray) and their brightness ([value](http://desktoppub.about.com/od/glossary/g/Value.htm) or luminance).

The HSV color wheel may be depicted as a cone or cylinder. Instead of Value, the color model may use Brightness, making it HSB (Photoshop uses HSB).

* Hue is expressed as a number from 0 to 360 degrees representing hues of red (starts at 0), yellow (starts at 60), green (starts at 120), cyan (starts at 180), blue (starts at 240), and magenta (starts at 300).
* Saturation is the amount of gray (0% to 100%) in the color.
* Value (or Brightness) works in conjunction with saturation and describes the brightness or intensity of the color from 0% to 100%.

Digital image is more powerful approach for flower recognition than traditional approach. So, in this flower identification system digital image of the flower to be recognized will be the main input which will be used by the system.

Objectives of this study

* To develop software for flower recognition using image processing techniques.
* Prototype implementation and testing of proposed software.

Review of literature

Das et al. (1999) gave us a technique of flower patent image indexing which takes a help of flower color and their structural location’s domain knowledge. Green, black, grey and brown are the color that hardly appears on flower region, also the images have background color that is usually visible and affect the software’s working. Therefore an automation withdrawal software was used which separated the real flower from the background as only color’s have the flower were used rather than complete colors of digital image. But using color information alone without the use of shape and area is not a sufficient approach.

Hong et al. (2004) gave an approach based on region of interest. Region of interest i.e. ROI extracts the flower image from the whole image of flower including background. In addition to the approach of Das et al. color clustering method was included for segmentation. The scattering of colors of the flower region is represented by color histogram, along with which for exact comparing two shape features were included.

Aleya and Samanta (2013) proposed that trading of flowers, production of seeds, nursery and potted plants and segmentation of useful oil from flowers are included in floriculture. Firstly, the image gathered from different regions for the process is viewed and enhanced. Secondly, the image disporting is done to get the targeted parts of flower. Lastly, final analysis of the detected region is done. The histogram approach is used to detect the disease taking the help of stem value. Finally the agricultural experts are looking out for consultative approach.

Pornpanomchai et al. (2011) has an objective to build software for computers that uses leaf or flower to identify a plant. This software has 4 different modules i.e. acquisition of image, preprocessing of image, recognition of image and result display. In image acquisition the flower image or leaf are captured on a background of white paper. Different processing algorithms are applied on flower or leaf in image preprocessing part. In image recognition, 8 main features are used for extraction. Finally in display result, the results best matched are displayed.

Saitoh and Kaneko(2000) introduced an idea that uses digital images for automatically identifying the wild flowers in leaf or flower image. Firstly, a black sheet is placed below the leaf or flower and to segment the background from flower and leaf, a clustering algorithm named k-means is used. A neural network approach is used to separate the 17 features that include shape, colour and many other properties.

Apriyanti et al.(2013) proposed a system that uses flower images to identify different species of orchids. Different shape features are extracted using MSRM i.e. Maximal Similarity based on Region Merging. Background is segmented from the flower object with the help of MSRM. HSV colour features and SVM methods are also used.

Tiay et al.(2014) developed image processing for flower recognition system. Classification of flower is done on the basis of colour and edge characteristics. Histogram is used to derive green, hue, blue, red and saturation features. Classification is done using k-nearest neighbour algorithm. Approximately more than 80% of the result is accurate.

Zou and Nagy (2004) developed a model-based interactive flower recognition system based on the concept of Computer Assisted Visual Inter Active Recognition (CAVIAR). In the training process, each training image was interactively segmented in order to extract the flower regions. interactive process will repeat until the user accepts the recognition result. One major problem of this system is that too many user interactions have to be conducted to get high recognition accuracy.

Nilsback and Zisserman (2006) investigated to what extent ‘bag of visual words’ models can be used to distinguish categories which have significant visual similarity. They developed and optimize a nearest neighbour classifier architecture, which is evaluated on a very challenging database of flower images. The flower categories are chosen to be indistinguishable on colour alone, and have considerable variation in shape, scale, and viewpoint. they demonstrated that by developing a visual vocabulary that explicitly represents the various aspects (colour, shape, and texture) that distinguish one flower from another, we can overcome the ambiguities that exist between flower categories. The novelty lies in the vocabulary used for each aspect, and how these vocabularies are combined into a final classifier.

Vibhute *et al* (2012) conducted a survey that intended to focus on the application of image processing in agriculture field such as imaging techniques, weed detection and fruit grading. The analysis of the parameters has proved to be accurate and less time consuming as compared to traditional methods. Image processing has been proved to be effective tool for analysis in various fields and applications. Many times expert advice may not be affordable, majority times the availability of expert and their services may consume time. Image processing along with availability of communication network can change the situation of getting the expert advice well within time and at affordable cost since image processing was the effective tool for analysis of parameters. Application of image processing can improve decision making for vegetation measurement, irrigation, fruit sorting, etc.

Fukuda et al (2008) developed a flower image retrieval system by combining multiple classifiers using fuzzy c-means clustering algorithm. In their system, flowers were classified into three categories of different structures: gamopetalous flowers, many-petaled flowers, and singlepetaled flowers. For each structure, a classifier with specific feature set was constructed. Fuzzy cmeans clustering algorithm was then used to determine the degree of membership of each image to each structure. The overall similarity is a linear combination of each individual similarity computed for each classifier with the weight being the degree of membership. Experimental results have shown that the multiple-classifier approach outperforms any single-classifier approach. However, it is too rough a classification mechanism to classify flowers into three different categories according to the number of petals.

Kim et at(2009) shows that conventional flower or leaf recognition studies have some restrictions and limitations. These include a sharp drop in recognition rate due to the varying positions and number of relevant objects in the original object image. Hence, this paper suggests and implements a mobile-based flower recognition system using Difference Image Entropy (DIE) and contour features of the flower from the original image with multiflower objects. In system framework includes 1) WiBro Net.- based transmission and designation module of the relevant flower object by drawing the flower region of the user's interest, 2) contour feature extraction module, and 3) DIE-based flower recognition module.

Holmes et al (2009) explained that supervised learning can be used to segment/identify regions of interest in images using both color and morphological information. The algorithms are also showing promise in other domains. The success of the method depends heavily on the use of color, the relative homogeneity of object appearance and on interactivity. As is often the case in segmentation, an algorithm specially tailored to the application works better than using broader methods that work passably well on any problem. Our main innovation is the interactive feature extraction from color images.

Singh and Vij (2012) proved that effective image enhancement can be done through histogram processing. Image enhancement is means to improvement of an image appearance by increasing or decreasing dominance of a feature or ambiguity between different regions of image. Four histogram techniques were tested on a low contrast image.

Note that the previous researchers extracted color or shape features from the whole image or flower region without specifically using characteristics of stamen/ pistil area. Thus software was proposed which extracts not only the shape and color feature from the whole flower region but also from stamen/ pistil area which will help us to identify flower more precisely.

Material and Methodology

The traditional approach for recognition of flowers is not that easy for flowers that are not part of our day to day life. To recognize a flower a person needs to have knowledge of all types of flowers including different colours, shapes and many other factors of flowers along with that he/she should also have experience in flower taxonomy. It becomes quite difficult , hectic and time consuming for a person to recognize a flower even with help from books or websites without any knowledge of keywords of flowers. People face a lot of problems in flower recognition despite of many software systems developed to help them. The search for flowers is restricted to keywords which makes it difficult for user without any knowledge in this field. Also the software systems developed are all recognizing flowers on the basis of its shape and colour features but in most of the flowers the petal/ rays or stamen/pistil area acts as different entities and should be considered as such. This problem of flower detection is solved in the proposed software.

Proposed software

In this paper we present freeware software for recognizing flower images taken by digital cameras with white background. The proposed system takes such images as input then it extracts the H value of the flower from the HSV colour model which represents the colour of the input flower. Thus colour feature of input flower is compared with the flowers existing in the proposed software and flowers with similar colour value are extracted and saved. Once the colour feature is matched colour image segmentation is done to extract the shape of the flower. The largest contour is picked as shape of the flower. A contour is the connected components with same colour. The shape of the flower is compared with the flowers extracted having same colour as the input flower. The flowers with match factor value less than particular threshold value are extracted as they have relevant shapes to the input flower. Mostly flowers are recognized till this phase. Now input flower’s relative stamen/ pistil area is calculated. After comparison of this area the flower is mostly identified. If not an interactive interface allows user to select the flower similar to input flower hence giving us 100% accurate result. The experiment is conducted on a database consisting of 24 species that in addition to colour and shape features when feature of stamen/pistil area is added result becomes more accurate.

In present study, a software developer needs to be aware of all the ways with which he/ she can fulfil the requirements of the software so that the software designed should be efficient. Also the designed software should be robust, user friendly and without errors. Hence it is necessary for a developer to follow a systematic approach while implementing various steps for the systems software development.

For the systematic development of proposed system a concept of software engineering has been used as given below:

**3.1 SDLC (Software Development Life Cycle)**

All the phases of SDLC are followed by during the development of every project. I also followed this cycle for the development of my project. The various phases are as:

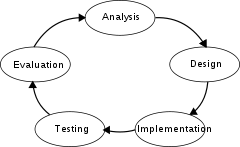


Figure 3.1: Stages of SDLC

**3.1.1 FEASIBILITY ANALYSIS**

**Feasibility Analysis: -** Not everything imaginable is feasible, not even in software, it may appear to outsiders. On the contrary, software feasibility has four solid dimensions:

**3.1.2.1 Technical Feasibility:** Yes, this product name as Sitting Plan Automation is technically feasible, as it is within the state of art and any defects can be further reduced to a level matching the needs of application. The organization have all the resources needed to succeed as they require an expert who can operate the computer well and has knowledge of all the controls and can handles the exceptions quite efficiently. The software’s used in these projects are open source and free of cost, easily available.

**3.1.2.2 Feasibility Analysis:** The feasibility analysis is designed to determine whether or not, given the project environment, a project will be successful. It is a determination as to the likelihood of success and a description of how that determination was achieved. It includes various elements like:

**3.1.2.3 Technology and system feasibility:** The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new website will perform adequately on a system or not. Technological feasibility is carried out to determine whether the company has the capability, in terms of software, hardware, personnel and expertise, to handle the completion of the project and considering this point we can say that the project is feasible because not much of extraordinary or expensive software’s are required and no extra hardware or workforce is needed. All the stuff available in the organization and some seminar which I have attended is enough to complete the project.

**3.1.2.4 Economic feasibility:** Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. It is important to identify cost and benefit factors, which can be categorized as follows:

1. Development costs.

2. Operating costs.

This is an analysis of the costs to be incurred in the system and the benefits derivable out of the system. Considering this factor also we can say that the project is feasible because all the software’s that we used incurred negligible expense.

### 3.1.2.5 Operational feasibility: Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition. The project also complies with this sort of feasibility as it would solve the problems of database connectivity and lack of an interactive front in the application. Moreover it would present a good application to the viewer. It is easy to use and operations and functionality can be easily understood by a professional. So, I can say “SITTING PLAN AUTOMATION “is operationally feasible.

**3.1.2 Software Requirement Specification (SRS)**

Once the captured data is analyzed these are put together in the form of a software requirement Specification document (SRS) or a system requirement specification (SRS) document. This document serves as a blueprint for the design or development teams to start building the solution on. It serves as a technical compendium of all the stakeholders’ needs including user requirements, system requirements, user interface and operational requirements.

**3.1.4 Design**

Describes desired features and operations in detail, including screen layouts, business rules, process, diagrams, pseudo code and other documentation. A prototype should be developed during the logical design phase if possible. The detailed design phase modifies the logical design and produces a final detailed design, which includes technology

Choices, specifies a system architecture, meets all system goals for performance, and still has all of the application functionality and behavior specified in the logical design. The design phase has two parts:

* Functional, detailed specification of all the system elements (Data, process, inputs and outputs.)
* Technical, detailed specifications of all the system elements (programs, file, network, system software etc.)

The Design stage describes how the proposed solution is to be developed. The solution design is specific to the systems technical environment and the tools to be used in constructing the system.

The results of this stage will be inputs to the Execute and Implement stages. Some useful tools for designing solutions are.

**3.1.4.2 DFDs**

The DFDs are also prepared into this phase. It specifies the graphical representation of project and after that it is converted into coding. A Data Flow Diagram (DFD) is a diagrammatic representation of the information flows within a system which showing how information enters and leaves the system, what changes the information and where information is stored. So simply we can say data flow diagram (DFD) is a graphical representation of the”flow” of data through an information system. In the year 1970s data-flow diagrams (DFDs) were introduced and popularized for structured analysis and design.

* **Purposes of DFDs**
* Freedom from committing to the technical implementation too early
* Gathering feedback information from user presentation
* Identify information requirements
* Understanding of the interrelationships of systems and subsystems
* **Uses of DFDs**
* Used to analyze the system to ensure that the design is complete
* Used to partition the system into programs
* Used for system documentation
* In designing a DFD the exact details of the process including issues such as timing, are not factors.

DFDs consist of four basic components that illustrate how data flows in a system: entity, process, data store, and data flow.

* **Symbols used for DFDs:**

**External Entity:** An external entity is a source or destination of a data flow which is outside the area of study. So it represents a person or a part of an organization which sends or receives data from the system.

* **Process:** A process shows a transformation or manipulation of data flows within the system.
* **Data Flow:** A data flow shows the flow of information from its source to its destination. It represents the exchange of data between processes, processes and data stores and processes and external entities.
* **Data Store:** A data store is a holding place for information within the system. It is represented by an open ended narrow rectangle. Normally data stores may be long term files such as sales ledgers or may be short term accumulations

**3.1.5 Coding**

Once the design is complete, most of the major decisions about the system have been made. The goal of the coding phase is to translate the design of the system into code in a given programming language. For a given design, the aim of this phase is to implement the design in the best possible manner. The coding phase affects both testing and maintenance profoundly. A well written code reduces the testing and maintenance effort. Since the testing and maintenance cost of software are much higher than the coding cost, the goal of coding should be to reduce the testing and maintenance effort.

**3.1.6 Testing**

Brings all the pieces together into a special testing environment, then checks for errors, bugs and interoperability. Testing is the process of evaluating a system or application, to check whether the application meets all requirements of the client and to detect the errors. Generally testing can be classified into static testing and dynamic testing. Again Dynamic Testing is classified into two types:

Structural Testing (or) white box

Functional Testing (or) Black Box testing.

* **Black Box Testing:** Black box testing involves looking at the specifications and does not require examining the code of a program. Tests that examine the observable behavior of software as evidenced by its outputs without referencing to internal functions is black box testing. It is not based on any knowledge of internal design or code and tests are based on requirements and functionality.
* **White box Testing:** Testing requires programming knowledge to know the internals of the code. Also it is time consuming. So only a developer can become a white box tester.
* **Unit Testing:** Soon after the program is corrected for syntax errors, the program has to be checked for logical errors, at the unit level. Programs may have simple user inputs or outputs thru screens or reports. The inputs must be validated for their format, data type, boundary conditions etc. Also, the elementary functionality of the program must be verified.
* **Integration Testing:** When all the individual program units are tested in the unit testing phase and all units are clear of any known bugs, the interfaces between those modules will be tested, to establish that they communicate to each other property via the specified APIs and thus they can be integrated into an application. The integration test may be performed by the independent testers or by the development team members.
* **System Testing:** After all the interfaces are tested between multiple modules, the whole set of software is tested to establish that all modules work together correctly as an application or system or package. This is again performed by independent testes. The testers will have to do the system testing as though they are the end users of the application. Systems testing include special testing methods like performance testing, interoperability testing, stability testing etc.

**3.1.7 Implementation**

The final stage of initial development, where the software is put into production and runs actual business. Project implementation process entails creation of a customizable framework that helps project managers to set up and manage project implementation stages. Customization of project implementation process framework lets leverage the use of management standards, policies and procedures and ensures that management expectations and plans for project implementation stages are properly outlined and applied.

When project implementation process is structured, customized and organized into consistent project implementation steps, all conditions required for creation of a responsive project management environment are met, and project manager can start implementing a project. If there are several projects to be implemented, project implementation

**1.3.6 Maintenance**

What happens during the rest of the software’s life: changes, correction, additions, and moves to a different computing platform and more. This is often the longest of the stages. Maintenance is the last stage in the system development life cycle and, consequently, is affected by everything that happens in the previous stages. Errors made during the analysis and design stages can significantly impact maintenance. More specifically, maintenance relies on the documentation created during the analysis and the system maintenance life cycle parallels the system development life cycle. Maintenance begins when the system is released and continues for the life of the system

* **Corrective maintenance:** The objective of corrective maintenance is to remove errors or bugs from the software, the procedures, the hardware, the network, the data structures, and the documentation. Corrective maintenance activities include both emergency repairs (fire fighting) and preventive (or corrective) repairs.
* **Adaptive maintenance:** The point of adaptive maintenance is to enhance the system by adding features, capabilities, and functions in response to new technology, upgrades, new requirements, or new problems. Note that adaptive maintenance is reactive. The idea is to fix the system when the general business climate, competition, growth, new technology, or new regulations make change necessary. The key to minimizing adaptive maintenance costs is to isolate system-dependent features.
* **Perfective maintenance:** The point of perfective maintenance is to enhance the system by improving efficiency, reliability, functionality, or maintainability, often in response to user or system personnel requests. Corrective and adaptive maintenance are reactive. Bugs are fixed as they are discovered. An upgrade to an operating system can necessitate a change to application software. Perfective maintenance, in contract, is proactive.
* **Preventive maintenance:** The objective of preventive maintenance is to anticipate problems and correct them before they occur. Files and databases must be updated, periodically reorganized, and regularly backed up. Control totals must be reset. New software releases must be installed. System performance monitoring is an important key to preventive maintenance. The idea is to conduct periodic audits and to run regular benchmark tests to determine if the system is continuing to perform to expectations. Both hardware and software are monitored to measure system load and system utilization. The information derived from performance monitoring provides an early warning of potential system problems and often initiates other forms of maintenance.
* **Managing maintenance:** Maintenance is expensive. The elements of a system often interact in unexpected ways, and ripple effects (unexpected bugs or new errors caused by a change intended to fix an initial problem) can be devastating.

**Chapter-3.2**

**Facilities Required For Proposed Work**

**3.2.1 Hardware Configuration:** The hardware used for the development of the project is given in below table.

Table 2 : Hardware configuration for SPA(Sitting Plan Automation)

|  |  |
| --- | --- |
| PROCESSOR: | Intel Core 2 Dual 2.13 Ghz |
| RAM: | 1 Gb DDR2 Ram |
| MONITOR: | 15” Color |
| HARD DISK: | 160 Gb |
| CDDRIVE: | Lg 52x |
| KEYBOARD: | Standard 102 Keys |
| MOUSE: | 3 Button Mouse |

**3.2.2 Software configuration and Tools:** The software used for the development of the project is mentioned below:

Table 3: Software configuration for Sitting Plan Automation

|  |  |
| --- | --- |
| OPERATING SYSTEM: | Windows XP professional |
| ENVIRONMENT: | ECLIPSE IDE 3.7 |
| BACK END | MySql |
| JAVA DEVELOPMENT KIT: | JDK 1.7 |
| LANGUAGE: | Advanced Java/ Html/ CSS, Java Script/Ajax |
| ARCHITECTURE | Sturts 2 |
| FRONT END | Java Server Pages |

**Chapter – 3.3**

**Introduction to the Technologies used in project**

**3.3.1 Introduction to Java**

JAVA was conceived by James Gosling, Patrick Naughton, Chris Warth, Ed Frank and Mike Sheridan at Sun Microsystems, Inc. in 1991. It took 18 months to develop first working version. This language was initially called “OAK” but was renamed JAVA in 1995.

Somewhat surprisingly, the impetus for Java was not the Internet. Instead, the primary motivation was the need for a platform-independent language that could be used to create software to be embedded in various consumer electronic devices, such as microwave ovens and remote controls. About the time that the details of Java were being worked out, a second ultimately more important, factor was emerging that played a crucial role in determining the future of the Java. This second force was, of course World Wide Web. With the advent of World Wide Web the need for platform independent applications was felt and since Java promised development of such applications, Java was propelled to the forefront of computer language design.

**3.3.1.1** **Scope of Java:** There are some areas of communication and information like mobile applications and embedded etc. where java has proved it very useful. Java has a chance to grow more and achieve a lead in these areas.

**3.3.1.2 Advantages of Java**

* The advantages of Java are as follows:
* Java is easy to learn.
* Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages.
* Java is object-oriented.
* This allows you to create modular programs and reusable code.
* Java is platform-independent.
* One of the most significant advantages of Java is its ability to move easily from one computer system to another. The ability to run the same program on many different systems is crucial to World Wide Web software, and Java succeeds at this by being platform-independent at both the source and binary levels.

**3.3.1.3 Features of Java Language**

* **Secure**: Whenever we download a program on our computers from Internet we expose our system to viral infections. Prior to Java people did not frequently download executable programs from the Internet and if they did so scanned their computers for possible viral infections. Apart from viruses another type of malicious program that might be downloaded from the Internet may collect private information like passwords or other personal information by reaching the local file systems of the computer. Java answers both these concerns by providing a firewall between a networked program and your computer. Java provides this protection by limiting its applications to Java run time environment and not allowing it to access other parts of the computer.
* **Portable**: The ability of Java to provide platform independent applications greatly solves this problem, thus making Java programs portable.
* **Multithreaded**: Java supports multithreaded programming, which allows you to write programs that do many things simultaneously. The Java run time system comes with an elegant and sophisticated solution for multiprocessor synchronization.
* **Architecture-Neutral**: The main objective in the designing of Java was to create architecture-neutral programs. The designers wanted to create programs that are written once and could be executed anytime, anywhere. To achieve this objective the designers had to take many hard decisions in the design of Java language and Java Virtual Machine.
* **Distributed:** Java is designed for the distributed environment of the Internet. The Java was developed to handle the environments of the Internet in an elegant and efficient manner.

**3.3.1.4 Java Byte code**

Byte code is a highly optimized set of instructions designed to be executed by the Java run-time system, which is called the Java Virtual Machine. That is in its standard form Java Virtual Machine is an interpreter for Byte code.

**3.3.1.5JVM**Java applications can run in the context of a Java Virtual Machine. A Java Virtual Machine is a runtime environment that executes a special from of Java binary code called byte code and converts this into appropriates calls for the local operating system. The JVM byte code is what makes application code that is write once, run anywhere possible. As long as a JVM is available for a hardware-based platform Java application byte code will run on the hardware platform. The Java API is a large collection of ready-made software components that provide many useful capabilities such as Graphical User Interface [GUI] widgets and so on. The Java API is grouped into packages of related components.

**3.3.1.6 JDBC**

In 1996, Java Soft released its first version of the JDBC kit. JDBC stands for Java database Connectivity. This is actually an API, which consists of a set of Java classes, interfaces and exceptions bound to a specification. JDBC driver vendors and developers who use JDBC must adhere to this specification when developing applications. Applications developed with Java JDBC are platform and database vendor independent i.e. the same Java program can simply by changing the JDBC middleware. An idea similar to Microsoft’s Open Database Connectivity [ODBC] is the JDBC underlay. JDBC [and ODBC] are based on the X/Open SQL Call level Interface [CLI], which is the basis for the ODBC standard for database connectivity. Programs written using the JDBC API communicate with a JDBC driver manager, which uses driver specifically loaded to communicate with the DB engine.

**3.3.2 Eclipse**

An Eclipse module is a Java archive file which contains Java classes written to interact with the [Eclipse APIs](file:///C:\Users\dell\Desktop\about-open-apis.html). A module identifies itself as a module by an entry in its MANIFEST.MF file. Eclipse modules are packaged as NBM files (**.nbm** extension) for non-installer distribution, usually via the Plug-in manager under the Tools menu. Eclipse modules are written with one of two aims in mind:

* **Extending the IDE.** You can very easily generate skeleton code for extending the IDE's functionality with new features. For example, you can use the skeleton code to write modules that make your favorite cutting-edge technologies available to the Eclipse IDE. Or, if you miss some functionality in the IDE, you can add it yourself, by using the skeleton code to write a module that provides the desired functionality.
* **Building a rich-client application.** You can use the core of the IDE as a platform on top of which you develop standalone desktop applications. The core of the IDE is a separate product called the [Eclipse Platform](file:///C:\Users\dell\Desktop\about-netbeans-platform.html). By basing your application on the Eclipse Platform, you can save a lot of development time, because you can reuse the platform's existing features such as menus, toolbars, and windowing systems.

**3.3.3 Packages**

To make types easier to find and to use, to avoid naming conflicts, and to control access, programmers bundle groups of related types into packages. A package is a collection of related types providing access protection and namespace management. Note that types refer to classes, interfaces, enums and annotations. The types that are part of the Java platform are members of various packages that bundle classes by function: fundamental classes are in java. Lang, classes for reading and writing (input and output) are in java.io, and so on. You can put your types in packages, too.You should bundle these classes and the interface in a package for several reasons: You and other programmers can easily determine that these types are related. You and other programmers know where to find types that provide graphics-related functions.

**3.3.3 JSP(Java Server Pages)**

JSP technology of Java Server Pages technology is an extension of Java programming technology. JSP includes a scripting language that is Java based. A JSP page on compilation generates a servlet. Web applications that are developed by using JSP demonstrate platform and Web Server allows Java to be embedded directly into an HTML page by using special tags. Although JSP clearly defines the work profiles of a web designer and a web developer or programmer, facilitating easy application development, there some instances that require explicit use of servlets. File type format support to extend web servers functionality:   
To add support for a new file format, a servlet can easily be mapped to the specified format, as is the case in the compilation cycle of a JSP page. The servlet is simply mapped to the JSP files. As a result, files with jsp extension are parsed and complied to generate servlets.

**3.4 Java Script**

Java script is a scripting language. To develop interactive Web sites, you need to create a Web interface for accepting data from users. The data accepted can be stored for further processing and validation. So we use the Java Script. You will also be able to write Java Script code to validate the data entered by users on the Web sites, handle run-time and execution errors in Web documents, and communicate with the users by displaying messages.

**3.3.5 Application Server: Apache Tomcat**

Tomcat is an application server from the Apache Software Foundation that executes Java servlets and renders Web pages that include Java Server Page coding.

**3.3.6 Struts**

The basic purpose of the Java Servlets in struts is to handle requests made by the client or by web browsers. In struts JavaServerPages (JSP) are used to design the dynamic web pages. In struts, servlets helps to route request which has been made by the web browsers to the appropriate ServerPage. The use of servlet as a router helps to make the web applications easier to design, create, and maintain. Struts is purely based on the Model- View- Controller (MVC) design pattern. It is one of the best and most well developed design patterns in use. By using the MVC architecture we break the processing in three sections named Model, the View, and the Controller.The **model** contains the business logic and interacts with the persistence storage to store, retrieve and manipulate data.The **view** is responsible for displaying the results back to the user. In Struts the view layer is implemented using JSP.The **controller** handles the entire request from the user and selects the appropriate view to return. In Struts the controller's job is done by the Action Servlet.

**3.3.7 HTML**

Hypertext Markup Language was developed by a consortium of organizations, the World Wide Web consortium (W3C). HTML is a segment used to create Web pages. Web pages are documents that you view on the Web. Web pages are stored as files with the extension .htm or .html. HTML is a web based language. It is very easy to learn and use.

**3.3.8 CSS**

CSS stands for Cascading Style Sheet. It defines a way how the information is presented by all the browsers on the web. A style sheet is a set of rules that controls the formatting of HTML elements on one or more web pages. Thus, the appearance of a Web page can be changed by changing the style sheet associated with it. There is no need to make detailed changes within the Web page to change how it looks. As one style sheet can be used for a whole web site, it normally means that the overall size of the website is smaller and downloads required for each page can be decreased by up to 40%.

**3.3.9 Back-End MySQL**

MySQL is a powerful Relational Database Management System (RDBMS) which we will use to learn the basic principles of database and data manipulation using Structured Query Language (SQL) statements. SQL is a database language that is used to retrieve, insert, delete and update stored data. This is achieved by constructing conditional statements that conform to a specific syntax (i.e. the strict order required of elements for a statement to work).