

**RECKONING ATTAINABLE INTENT FOR DIKE WITH
UNCHARTED AGITATION**

at

**Sathyabama Institute of Science and Technology
(Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of Bachelor
of Engineering Degree in Computer Science and Engineering

By

**Kesava Muthukumar
REG. NO. 38110249**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

MARCH 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY **(DEEMED TO BE UNIVERSITY)**

Accredited with Grade “A” by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI– 600119
www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of Kesava Muthukumar (38110249) who carried out the project entitled “Reckoning Attainable Intent for Dike with uncharted Agitation” under my supervision from June 2021 to November 2021.

Internal Guide

Ms.Jayanthi.S

Head of the Department

Dr.S.Vigneshwari ,ME, Ph.D, and Dr.L.Lakshmanan ME,PH.D

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **Kesava Muthukumar** hereby declare that the project report entitled “**Reckoning Attainable Intent for Dike with uncharted Agitation**” done by me under the guidance of **MS.Jayanthi.S** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE:10-04-2022

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGE,. MENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean, School of Computing, Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.S.Jayanthi .M.E., Ph.D.** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT:

In early stage there were different ways to produce the energy. There are different energy sources such as wind energy, mechanical energy, kinetic energy, gravitational energy, thermal energy, chemical energy, electrical energy, light energy, radiant energy, sound energy, nuclear energy, wind energy so on. Even we have reaped many types of energy source, in this all types; some of the energy is not authorized to commercial people. The easiest eco-friendly energy both industrial and commercial purpose use solar energy. The energy can be used directly to heat, and light homes or it can be converted into electricity using solar energy technologies like solar panels. People are increasingly investing in solar energy to save on out-of-pocket costs and get rid of dangerous and expensive power lines. In past if a person wants to purchase solar panel they do not have any prior knowledge about to buy the panel because on that stage the solar panel was made only in photovoltaic system and that time the price dramatically increases. As the technology develops the solar panel types are also increased, by some of the people don't know the clear idea to purchase which panel, energy consumption etc. we proposed in the future model if a person not get knowledge about the solar panel by this application the people get idea about it.

CONTENTS

1. INTRODUCTION

- 1.1. INTRODUCTION TO PROJECT
- 1.2. PURPOSE OF THE PROJECT

2. SYSTEM ANALYSIS

- 2.1. INTRODUCTION
- 2.2. ANALYSIS MODEL
- 2.3. SDLC PHASES
- 2.4. HARDWARE & SOFTWARE REQUIREMENT
- 2.5. INPUT AND OUTPUT
- 2.6. LIMITATIONS
- 2.7. EXISTING SYSTEM
- 2.8. SOLUTION OF THESE PROBLEMS IN PROPOSED SYSTEM

3. FEASIBILITY REPORT

- 3.1 TECHNICAL FEASIBILITY
- 3.2 OPERATIONAL FEASIBILITY
- 3.3 ECONOMIC FEASIBILITY

4. SOFTWARE REQUIREMENT SPECIFICATIONS

- 4.1. FUNCTIONAL REQUIREMENTS
- 4.2. NON-FUNCTIONAL REQUIREMENTS
- 4.3. PERFORMANCE REQUIREMENTS

5. SYSTEM DEVELOPEMENT ENVIRONMENT

- 5.1 INTRODUCTION TO JAVA
- 5.2 SERVLETS,JSP
- 5.3 JDBC
- 5.4 HTML,JAVASCRIPT
- 5.5 FRAMEWORKS

6. SYSTEM DESIGN

- 6.1. INTRODUCTION
- 6.2. NORMALIZATION
- 6.3. SYSTEM ARCHITECTURE
- 6.4. E-R DIAGRAM
- 6.5. FLOW DIAGRAM
- 6.6. DFD SYMBOLS
- 6.7. ACTIVITY DIAGRAM
- 6.8. USE CASE DIAGRAM
- 6.9. SEQUENCE DIAGRAM
- 6.10 CLASS DIAGRAM

7. OUTPUT SCREENS

8. CODING

9. SYSTEM TESTING AND IMPLEMENTATION

- 9.1. INTRODUCTION
- 9.2. STRATEGIC APPROACH OF SOFTWARE TESTING
- 9.3. UNIT TESTING
- 9.4. TEST SCREEN SHOT

10. SYSTEM SECURITY

- 10.1. INTRODUCTION
- 10.2. SECURITY IN SOFTWARE

11. CONCLUSION& FUTURE ENHANCEMENT

CHAPTER-1

1.1 INTRODUCTION TO PROJECT

In early stage people are aware of solar energy by commercial people are not affordable due to high cost of price. In upcoming world the solar panel types are classified into many segments, price is increase or decrease according to panel types. On that stage also the people have no knowledge about what panel is used for their purpose and also the waste their money. By this situation we want to the change people mind to give awareness about importance of energy through this application. By this application the people get know about the panel types and the importance also their features, so people can easily understand it. By some of the people can check with cost wise, this feature also available in this application. In this feature can be done with concept of feature selection method, in this method is used to give the relevant information to the user. In energy preserving scheme selection approach to predict the material properties through machine learning and computational methods. In this application, we apply algorithms for panel prediction through user data's; it is a crucial step of data driven machine learning approach where the panels and the related information on the datasets. When the user give the specific data as input and it process to get relevant information as output.

1.2 PURPOSE OF THE SYSTEM

- ✓ To predict the data according to the customer budget.
- ✓ Image verification can be done.
- ✓

CHAPTER-2

SYSTEM ANALYSIS

2.1 INTRODUCTION

In early stage people are aware of solar energy by commercial people are not affordable due to high cost of price. In upcoming world the solar panel types are classified into many segments, price is increase or decrease according to panel types. On that stage also the people have no knowledge about what panel is used for their purpose and also the waste their money. By this situation we want to the change people mind to give awareness about importance of energy through this application. By this application the people get know about the panel types and the importance also their features, so people can easily understand it. By some of the people can check with cost wise, this feature also available in this application. In this feature can be done with concept of feature selection method, in this method is used to give the relevant information to the user. In energy preserving scheme selection approach to predict the material properties through machine learning and computational methods. In this application, we apply algorithms for panel prediction through user data's; it is a crucial step of data driven machine learning approach where the panels and the related information on the datasets. When the user give the specific data as input and it process to get relevant information as output.

2.2 ANALYSIS

SOFTWARE DEVELOPMENT LIFE CYCLE

INTRODUCTION:

The System Development Lifecycle framework is designed to outline a complete development and implementation process suitable for developing complex applications. SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

- Business – legislation regulatory requirements, policy, SOP's, guidelines etc.
- Process – how the business is implemented
- Data – the core business data elements collected for the business
- Application – the gate to the business collecting
- Infrastructure- the servers, network, workstations, etc.

2.3 SDLC Phases:



Stage 1: Scheduling and Requisite investigation:

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational, and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Stage 2: Significant necessities:

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is

done through .SRS. . Software Requirement Specification document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Scheming the product design:

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in the SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

Stage 4: Structure or Mounting the Product:

In this stage of SDLC the actual development starts and the product are built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers have to follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java, and PHP are used for coding.

Stage 5: Testing the Product:

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product, where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 6: Consumption in the Market and Safeguarding :

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometime product deployment happens in stages as per the organizations.

Business strategy. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

The product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

2.4 HARDWARE AND SOFTWARE REQUIREMENTS

Developing Kit			
	Processor	RAM	Disk Space
Eclipse	Computer with a 2.6GHz processor or higher	2GB	Minimum 20 GB
Database			
MySql 5.0	Intel Pentium processor at 2.6GHz or faster	Minimum 512 MB Physical Memory; 1 GB Recommended	Minimum 20 GB
HeidiSQL 8.3	Intel Pentium processor at 2.6GHz or faster	Minimum 512 MB Physical Memory; 1 GB Recommended	Minimum 20 GB

Software Requirements:

- **Front end** : core java, css, js, servlet
- **Web application** : J2ee Frameworks, Hibernate
- **Back end** : Mysql 5.1

OVERVIEW OF SOFTWARE ENGINEERING:

Software is more than just a program code. A program is an executable code, which serves some computational purpose. Software is considered to be collection of executable programming code, associated libraries and documentations. Software, when made for a specific requirement is called **software product**.

Engineering on the other hand, is all about developing products, using well-defined, scientific principles and methods.



Software engineering is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures. The outcome of software engineering is an efficient and reliable software product.

Definitions

IEEE defines software engineering as:

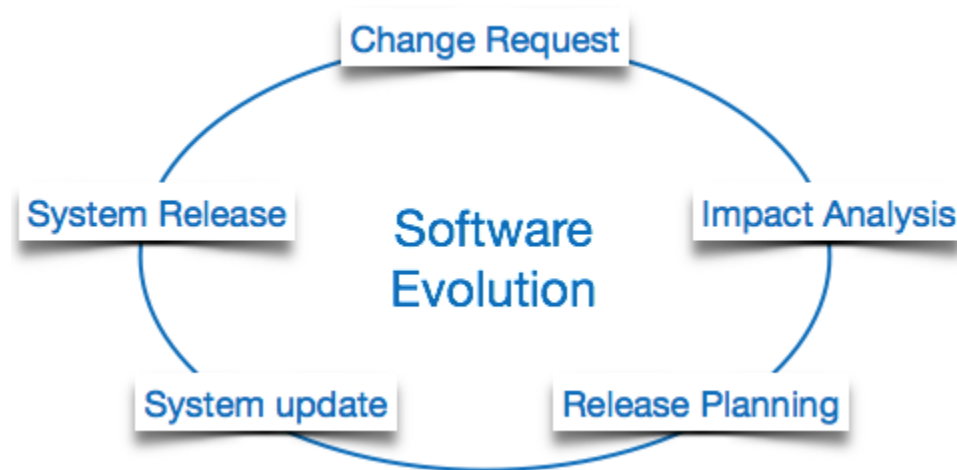
- (1) The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software; that is, the application of engineering to software.
- (2) The study of approaches as in the above statement.

Fritz Bauer, a German computer scientist, defines software engineering as:

Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and work efficiently on real machines.

Software Evolution

The process of developing a software product using software engineering principles and methods is referred to as **software evolution**. This includes the initial development of software and its maintenance and updates, till desired software product is developed, which satisfies the expected requirements.



Evolution starts from the requirement gathering process. After which developers create a prototype of the intended software and show it to the users to get their feedback at the early stage of software product development. The users suggest changes, on which several consecutive updates and maintenance keep on changing too. This process changes to the original software, till the desired software is accomplished.

Even after the user has desired software in hand, the advancing technology and the changing requirements force the software product to change accordingly. Re-creating software from scratch and to go one-on-one with requirement is not feasible. The only feasible and economical solution is to update the existing software so that it matches the latest requirements.

Software Evolution Laws

Lehman has given laws for software evolution. He divided the software into three different categories:

- **S-type (static-type)** - This is a software, which works strictly according to defined specifications and solutions. The solution and the method to achieve it, both are immediately understood before coding. The s-type software is least subjected to changes

hence this is the simplest of all. For example, calculator program for mathematical computation.

- **P-type (practical-type)** - This is a software with a collection of procedures. This is defined by exactly what procedures can do. In this software, the specifications can be described but the solution is not obvious instantly. For example, gaming software.
- **E-type (embedded-type)** - This software works closely as the requirement of real-world environment. This software has a high degree of evolution as there are various changes in laws, taxes etc. in the real world situations. For example, Online trading software.

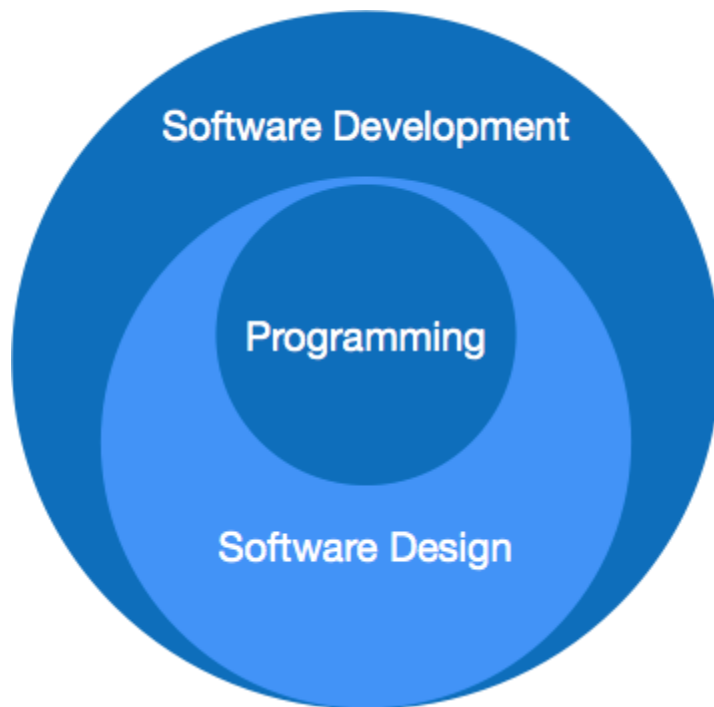
E-Type software evolution

Lehman has given eight laws for E-Type software evolution -

- **Continuing change** - An E-type software system must continue to adapt to the real world changes, else it becomes progressively less useful.
- **Increasing complexity** - As an E-type software system evolves, its complexity tends to increase unless work is done to maintain or reduce it.
- **Conservation of familiarity** - The familiarity with the software or the knowledge about how it was developed, why was it developed in that particular manner etc. must be retained at any cost, to implement the changes in the system.
- **Continuing growth**- In order for an E-type system intended to resolve some business problem, its size of implementing the changes grows according to the lifestyle changes of the business.
- **Reducing quality** - An E-type software system declines in quality unless rigorously maintained and adapted to a changing operational environment.
- **Feedback systems**- The E-type software systems constitute multi-loop, multi-level feedback systems and must be treated as such to be successfully modified or improved.
- **Self-regulation** - E-type system evolution processes are self-regulating with the distribution of product and process measures close to normal.
- **Organizational stability** - The average effective global activity rate in an evolving E-type system is invariant over the lifetime of the product.

Software Paradigms

Software paradigms refer to the methods and steps, which are taken while designing the software. There are many methods proposed and are in work today, but we need to see where in the software engineering these paradigms stand. These can be combined into various categories, though each of them is contained in one another:



Programming paradigm is a subset of Software design paradigm which is further a subset of Software development paradigm.

Software Development Paradigm

This Paradigm is known as software engineering paradigms where all the engineering concepts pertaining to the development of software are applied. It includes various researches and requirement gathering which helps the software product to build. It consists of –

- Requirement gathering
- Software design
- Programming

Software Design Paradigm

This paradigm is a part of Software Development and includes –

- Design
- Maintenance
- Programming

Programming Paradigm

This paradigm is related closely to programming aspect of software development. This includes

—

- Coding
- Testing
- Integration

Need of Software Engineering

The need of software engineering arises because of higher rate of change in user requirements and environment on which the software is working.

- **Large software** - It is easier to build a wall than to a house or building, likewise, as the size of software become large engineering has to step to give it a scientific process.
- **Scalability**- If the software process were not based on scientific and engineering concepts, it would be easier to re-create new software than to scale an existing one.
- **Cost**- As hardware industry has shown its skills and huge manufacturing has lower down the price of computer and electronic hardware. But the cost of software remains high if proper process is not adapted.
- **Dynamic Nature**- The always growing and adapting nature of software hugely depends upon the environment in which user works. If the nature of software is always changing, new enhancements need to be done in the existing one. This is where software engineering plays a good role.
- **Quality Management**- Better process of software development provides better and quality software product.

Characteristics of good software

A software product can be judged by what it offers and how well it can be used. This software must satisfy on the following grounds:

- Operational
- Transitional
- Maintenance

Well-engineered and crafted software is expected to have the following characteristics:

Operational

This tells us how well software works in operations. It can be measured on:

- Budget
- Usability
- Efficiency
- Correctness
- Functionality
- Dependability
- Security
- Safety

Transitional

This aspect is important when the software is moved from one platform to another:

- Portability
- Interoperability
- Reusability
- Adaptability

Maintenance

This aspect briefs about how well a software has the capabilities to maintain itself in the ever-changing environment:

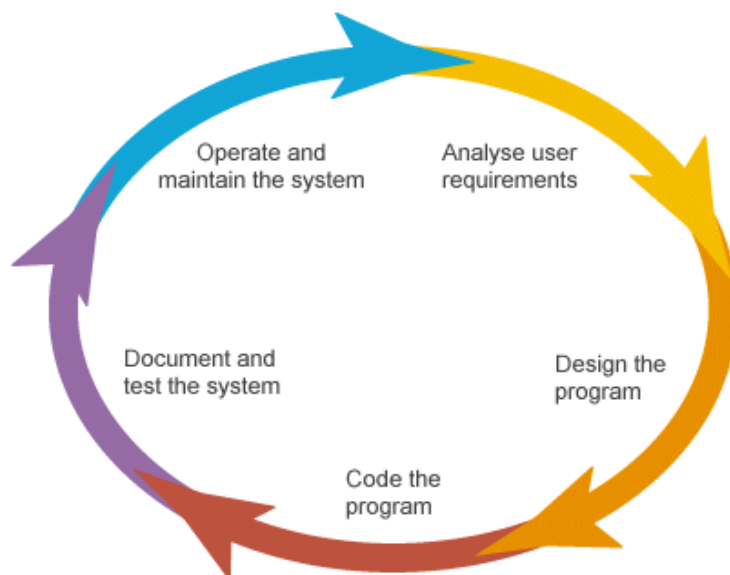
- Modularity
- Maintainability

- Flexibility
- Scalability

In short, Software engineering is a branch of computer science, which uses well-defined engineering concepts required to produce efficient, durable, scalable, in-budget and on-time software products.

SOFTWARE DEVELOPMENT LIFE CYCLE

The Software Development Life Cycle is a process that ensures good software is built. Each phase in the life cycle has its own process and deliverables that feed into the next phase. There are typically 5 phases starting with the analysis and requirements gathering and ending with the implementation. Let's look in greater detail at each phase:



Stage 1: Scheduling and Requisite Analysis

During the discovery phase our team conducts a detailed requirement analysis and creates a work-breakdown structure.

Stage 2: Scheming the product design

We identify the design and architecture of the project. SRS is the reference for product architects to come out with the best architecture for the product to be developed.

Stage 3: Structure or Mounting the Product

In this stage of SDLC the actual development starts and the product is built. Different high level programming languages such as C, C++, Pascal, Java, C# and PHP are used for coding.

Stage 4: Testing the Product

Testing is the last phase of the Software Development Life Cycle before the software is delivered to customers. During testing, experienced testers start to test the system against the requirements.

Stage 5: Consumption in the Market and Safeguarding

Once the product has been fully tested and no high priority issues remain in the software, it is time to deploy to production where customers can use the system.

2.5 INPUT AND OUTPUT

The major inputs and outputs and major functions of the system are follows:

Input:

- The entities such as authority and users must register the account for login. All the user details have been stored the data in our database for the purpose of identifying the entities.
- The main and sub authority uploads the necessary details to involve in the storing operation.

Output:

- Only the authorized data users can able to access the data.
- Prediction of outflow will be calculated at the given time.

2.5 INPUT DESIGN

- Input design is a part of overall system design. The main objective during the input design as given below.
- Input States: User can maintain a database in mysql server or sql server for his/her business requirement.
- Input Media:
At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to:
- In this section user can give the input for storage location and get the output from admin side.

2.6 LIMITATIONS

- ✓ The Dam water levels are measured manually.
- ✓ Requires lots of human efforts to measure and update.
- ✓ Decisions making is quite difficult.
- ✓ Communicating authorities at hard times is difficult.

2.7 PROBLEMS IN EXISTING SYSTEM:

In the existing stage, when the some of the people only educated, more no off are not educated so and that time they have no prior knowledge about panel to save the energy, by some people can buy the panel on that time the solar panel cost is in increasing rate why because then the panel one type is photovoltaic cell type panel. On upcoming world, then panel type increases and the price of the panel increase and decrease according to the panel type. Then the population increases and also increases the consumption of the current, by that stage people get knowledge about to buy a panel, the dealers or vendors get deceiving they sold low range output panel are sold at high cost for their profit because the people don't know the prior knowledge about types of panel and the features (monthly energy saving, etc...). This mentally known by the vendors for

their profit they sold for higher cost. By this people get wasting their money for their low quality product.

2.8 PROPOSED SYSTEM

We proposed with logical implementation in using the Machine Learning algorithm is comes under Feature method technique. Feature selection enables in the Artificial intelligence algorithm to train faster. It reduces the complexity of a model and makes it easier to interrupt. It improves the accuracy of a model if the right subset is chosen. Feature selection is the process of selecting a subset of the most relevant features from the given set of feature in a group of data example by discarding irrelevant, redundant, and noisy features, either based on or not based on the corresponding labels Person correlation coefficient (PCC) also known as Pearson's r is a measure of linear correlation between the set of data's and give relevant data as a output. Thus the algorithm are essentially normalized measurement of the covariance, such that the result always has a value between -1 and 1.by this algorithm is applied in our application people is to gain the knowledge about what should they buy in his budget. If they apply this budget and the algorithm get the relevant information's about the coated budget. By this information the user can have multiple options of products then the user can choose which type of product is suitable for this area i.e. commercial or private sector.

Advantage of Proposed System:

- Solar energy doesn't require any outside supply to work, so its maintenance and energy production costs are practically zero. The only cost associated with the use of solar energy is the manufacture and installation of the components.
- The production of solar energy doesn't generate noise pollution, which is an important aspect to take into account for installations in urban areas.
- It also doesn't generate any waste because it doesn't need maintenance and its lifetime is far longer than other energy-production systems.
- The production of clean energy from the sun significantly reduces costs, because it is an inexhaustible source of energy, so the people can save there money.

CHAPTER-3

FEASIBILITY REPORT

- User can be able to create the user id and it can schedule with proper alignment
- User can view the shares and can able to do trading online securely.
- Every broker can register in our application and do trading operation by themselves.

3.1 Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipments have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a browser based user interface for audit workflow. Thus, it provides an easy access to the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the rules specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

3.2 Operational Feasibility

The analyst considers the extent the proposed system will fulfill his departments. That is, whether the proposed system covers all aspects of the working system and whether it has considerable improvements. We have found that the proposed “Secure transaction” will certainly have considerable improvements over the existing system.

3.3 Economic Feasibility

The proposed system is economically feasible because the cost involved in purchasing the hardware and the software is within approachable. Working in this system need not require a highly qualified professional. The operating-environment costs are marginal. The less time involved also helped in its economic feasibility.

CHAPTER-4

SOFTWARE REQUIREMENT SPECIFICATION

4.1 INTRODUCTION

The purpose of this document is to present a detailed description of the Web application system. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the Regional Historical Society for its approval.

PURPOSE

The purpose of this Software Requirement Specification (SRS) is to help the project. It is provided with some requirements which are used in the Transaction Mercator System. All parts; design, coding and testing will be prepared with helping of SRS. The purpose of this document is to detail the requirements placed on the Transaction Mercator System and serves as a contract between the customer and the developers as to what is to be expected of the stock exchange, and how the components of the system are working with each other with external systems.

This document will be checked by the group member's supervisor and it will corrected by members if supervisor orders.

DEVELOPERS RESPONSIBILITIES OVERVIEW:

The developer is responsible for:

- Developing the system, which meets the SRS and solving all the requirements of the system?
- Demonstrating the system and installing the system at client's location after the acceptance testing is successful.
- Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.
- Conducting any user training that might be needed for using the system.
- Maintaining the system for a period of one year after installation.

4.1 FUNCTIONAL REQUIREMENTS:

- Following is a list of functionalities of the browsing enabled system.
- An Activity with a UI that allows you to browser settings. Provide a second Activity that allows users to access the share with permission from the administrator. Handle activity lifecycle appropriately. A precondition for any points in this part of the grade is code that compiles and runs.
- Your application should allow a user to browse the shares, buy and sell the shares with specific metadata. The assignment requires you to create a UI for browsing and a UI for integrating the two.
- The Net beans provide a number of useful layout components, views, and tools that you may want to use to create your location browser. As with the final project, you should design your application to only use the buttons on the Key board and mouse as input. Your application should use the Key board, Mouse and keywords.

4.2 NON-FUNCTIONAL REQUIREMENTS:

- The system should be supported Net beans. The member should use the System browser. Each member should have a separate system.
- The system should ask the username and password to open the application. It doesn't permit to unregistered user to access the System.
- The system should have Role based System functions access. Approval Process has to be defined.
- The system should have Modular customization components so that they can be reused across the implementation.
- These are the mainly following:
 - Secure access of confidential data. 24 X 7 availability
 - Better component design to get better performance at peak time
 - Flexible service based architecture will be highly desirable for future extension

4.3 PERFORMANCE REQUIREMENTS

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the required specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

CHAPTER-5

SYSTEM DEVELOPEMENT ENVIRONMENT

5.1 INTRODUCTION TO JAVA

About Java:

Initially the language was called as “oak” but it was renamed as “Java” in 1995. The primary motivation of this language was the need for a platform-independent (i.e. Architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

- Java is a programmer’s language
- Java is cohesive and consistent

- Except for those constraints imposed by the Internet environment. Java gives the programmer, full control

Finally Java is for Internet Programming where C was to System Programming.

Importance of Java to the Internet

Java has had a profound effect on the Internet. This is because; Java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the server and the personal computer. They are passive information and Dynamic active programs. In the areas of Security and probability. But Java addresses these concerns and by doing so, have opened the door to an exciting new form of program called the Applet.

Applications and applets. An application is a program that runs on our Computer under the operating system on that computer. It is more or less like one creating, using C or C++.Java's ability to create Applets makes it important. An Applet is an application, designed to be transmitted over the Internet and executed by a Java-compatible web browser. An applet is actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can be reacted to the user input and dynamically change.

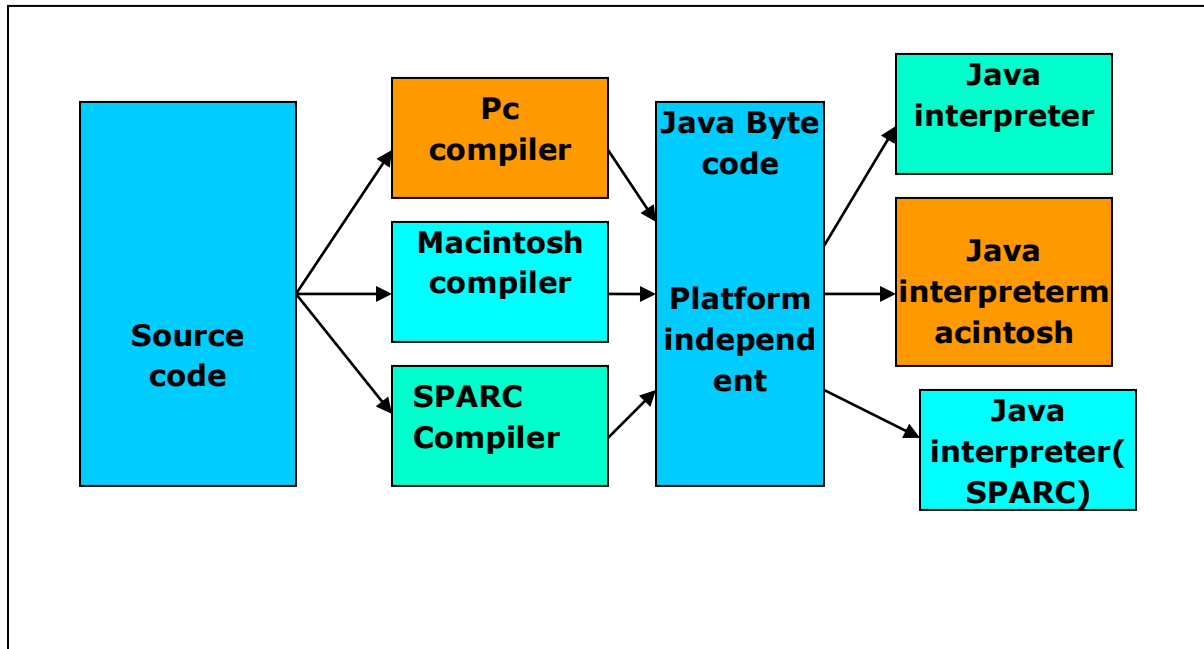
Java Architecture

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed for a machine in the same room or across the planet.

When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called a Java Virtual Machine (JVM). The JVM is supposed to be executed the byte code. The JVM is created for the overcoming the issue of probability. The

code is written and compiled for one machine and interpreted on all machines. This machine is called a Java Virtual Machine.

Compiling and interpreting Java source code.



During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. In reality this could be an Intel Pentium windows 95 or sun SPARCstation running Solaris or Apple Macintosh running system and all could receive code from any computer through the internet and run the Applets.

Simple:

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ Programmer. Learning Java will oriented features of C++. Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

Object oriented

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank state. One outcome of this was a clean, usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

Robust

The multi-platform environment of the web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs. Was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and runtime.

Java virtually eliminates the problems of memory management and deal location, which is completely automatic. In a well-written Java program, all run-time errors can and should be managed by your program.

5.2 SERVLETS/JSP

A Servlet is a generic server extension. Java classes that can be loaded dynamically to expand the functionality of a server. Servlets are commonly used with web servers. Where they can take the place CGI scripts.

A Servlet is similar to proprietary server extension, except that it runs inside a Java Virtual Machine (JVM) on the server, so it is safe and portable Servlets operate solely within the domain of the server.

Unlike CGI and Fast CGI, which use multiple processes to handle separate program or separate requests, separate threads within the web server process handle all servlets. This means that servlets are all efficient and scalable.

Servlets are portable; both across operating systems and also across web servers. Java Servlets offer the best possible platform for web application development.

Servlets are used as a replacement for CGI scripts on a web server; they can extend any sort of server, such as a mail server that allows servlets extend its functionality, perhaps by performing a virus scan on all attached documents or handling mail filtering tasks.

Servlets provide a Java-based solution used to address the problems currently associated with doing server-side programming including inextensible scripting solutions platform-specific API's and incomplete interface.

Servlets are objects that conform to a specific interface that can be plugged into a Java-based server. Servlets are to the server-side what applets are to the client-side. They differ from applets in that they are faceless objects (without graphics or a GUI component). They serve as platform independent, dynamically loadable, pluggable helper byte code objects on the server side that can be used to dynamically extend server-side functionality.

For example an HTTP servlet can be used to generate dynamic HTML content when you use servlets to do dynamic content you get the following advantages:

They're faster and cleaner than CGI scripts

They use a standard API (the servlet API)

They provide all the advantages of Java (run on a variety of servers without needing to be rewritten)

Attractiveness of servlets:

There are many features of servlets that make them easy and attractive to use these include:

- Easily configured using the GUI-based Admin tool
- Can be loaded and invoked from a local disk or remotely across the network.
- Can be linked together or chained, so that one servlet can call another servlet or several servlets in sequence.

- Can be called dynamically from within HTML, pages using server-side include-tags.
- Are secure-even when downloading across the network, the servlet security model and servlet and box protect your system from unfriendly behavior.

Advantages of the Servlet API

One of the great advantages of the servlet API is protocol independent. It assumes nothing about:

- The protocol being used to transmit on the net
- How it is loaded
- The server environment it will be running in

These quantities are important, because it allows the Servlet API to be embedded in many different kinds of servers.

There are other advantages to the servlet API as well these include:

- It's extensible-you can inherit all your functionality from the base classes made available to you
- Its small, simple, and easy to use.

Features of Servlets:

- Servlets are persistent. Servlet is loaded only by the web server and can maintain services between requests.
- Servlets are fast. Since servlets only need to be loaded once, they offer much better performance over their CGI counterparts.
- Servlets are platform independent.
- Servlets are extensible Java is a robust, object-oriented programming language, which easily can be extended to suit your needs.
- Servlets are secure
- Servlets are used with a variety of clients.

Every servlet must implement the `javax.servletinterface`. Most servlets implement it by extending one of two classes. `javax.servlet.GenericServlet` or `javax.servlet.http.HttpServlet`. A

protocol-independent servlet should subclass Generic-Servlet.while an Http servlet should subclass HttpServlet, which is itself a subclass of Generic-servlet with added HTTP-specific functionality.

Unlike a java program, a servlet does not have a main() method,Instead the server in the process of handling requests invoke certain methods of a servlet.Each time the server dispatches a request to a servlet, it invokes the servlets Service() method.

A generic servlet should override its service() method to handle requests as appropriate for the servlet.The service() accepts two parameters a request object and a response object. The request object tells the servlet about the request, while the response object is used to return a response

In Contrast. an Http servlet usually does not override the service () method. Instead, it overrides doGet () to handle GET requests and doPost () to handle Post requests. An Http servlet can override either or both of these modules the service() method of HttpServlet handles the setup and dispatching to all the doXXX() methods. Whichiswhy it usually should not be overridden

The remainders in the javax. servlet and Java. servlet. http. The package is largely supported class's.TheServletRequest and ServletResponse classes in Java. servlet is provided access to generic server requests and responses while HttpServletRequest and HttpServletResponse classes in javax.servlet provide access to generic server requests and responses while HttpServletRequest and HttpServletResponse in javax.servlet.http provide access a HTTP requests and responses. The javax.servlet.http provide contains an HttpSession class that provides built-in session tracking functionality and Cookie class that allows quick setup and processing Http Cookies.

Loading Servlets:

Servlets can be loaded from their places. From a directory that is in the CLASSPATH. The CLASSPATH of the JavaWebServer includes service root/classes/, which is where the system classes reside

From the <SERVICE_ROOT/servlets/directory. This is not in the server's classpath. A class loader is used to create servlet from this directory. New servlets can be added-existing servlets can be recompiled and the server will notice these changes. From a remote location. For this a code base like <http://nine.eng/classes/foo/> is required in addition to the servlet's class name. Refer to the admin GUI docs on servlet section to see how to set this up.

Loading Remote Servlets

Remote servlets can be loaded from:

- Configuring the admin Tool to setup automatic loading of remote servlets.
- Selection up server side include tags in .html files

Invoking Servlets

A servlet invoker is a servlet that invokes the “server” method of a named servlet. If the servlet is not loaded into the server, then the invoker first loads the servlet (either from local disk or from the network) and then invokes the “service” method. Also like applets, local servlets in the server can be identified by just the class name. In other words, if a servlet name is not absolute. It is treated as local.

A Client can Invoke Servlets in the Following Ways:

- The client can ask for a document that is served by the servlet.
- The client(browser) can invoke the servlet directly using a URL, once it has been mapped using the SERVLET ALIASES Section of the admin GUI
- The servlet can be invoked through server side include tags.
- The servlet can be invoked by placing it in the servlet/directory
- The servlet can be invoked by using it in a filter chain

The Servlet Life Cycle:-

The Servlet life cycle is one of the most exciting features of Servlets. This life cycle is a powerful hybrid of the life cycles used in CGI programming and lower-level NSAPI and ISAPI programming.

The Servlet life cycle allows servlet engines to address both the performance and resource problems of CGI and the security constraints of low level server API programming.

The servlet life cycle is highly flexible. Servers have significant leeway in how they choose to support servlets. The only hard and fast rule is that a servlet engine must conform to the following life cycle contract:

- Create and initialize the servlets
- Handle zero or more service for clients
- Destroy the servlet and then garbage collects it.

It's perfectly legal for a servlet to be loaded, created and initialized in its own JVM, only to be destroyed and garbage collected without handling any client request or after handling just one request.

The most common and most sensible life cycle implementations for HTTP servlets are: Single Java virtual machine and stateless persistence.

Init and Destroy:-

Just like Applets, servlets can define `init ()` and `destroy ()` methods. A servlet's `init (ServiceConfig)` method is called by the server immediately after the server constructs the servlet instance. Depending on the server and its configuration, this can be at any of these times:

- When the server starts
- When the servlet is first requested, just before the `service()` method is invoked
- At the request of the server administrator

The `init ()` method is typically used to perform servlet initialization, create or load objects that are used by the servlet in the handling of its request. In order to provide a new Servlet any information about itself and its environment, a server has to call a `servletsinit ()` method and pass an object that implements the `ServletConfig` interface.

This ServletConfig object supplies a servlet with information about its initialization parameters. These parameters are given to the servlets and are not associated with any single request. They can specify initial values, such as where a counter should begin counting, or default values, perhaps a template to use when not specified by the request,

The server calls a servlet destroy () method when the servlet is about to be unloaded. In the destroy () method, a Servlet should free any resources it has acquired that will not be garbage collected. The destroy () method also gives a servlet a chance to write out its unsaved. Cached information or any persistent information that should be read during the next call to init ().

USER AUTHORIZATION:

One way to perform session tracking is to leverage the information that comes with User authorization. When a web server restricts access to some of its resources to only those clients that log in using a recognized username and password. After the client logs in, the username is available to a servlet through getRemoteUser ().

When use the username to track the session. Once a user has logged in, the browser remembers her user name and resends the name and password as the user views new pages on the site. A servlet can identify the user through her username and they're by Track her session.

The biggest advantage of using user authorization to perform session tracking is that it's easy to implement. Simply tell the protect a set of pages, and use getRemoteUser() to identify each client. Another advantage is that the technique works even when the user accesses your site form or exits her browser before coming back.

The biggest disadvantage of user authorization is that it requires each user to register for an account and then log in each time the starts visiting your site. Most users will tolerate registering and logging in as a necessary evil when they are accessing sensitive information, but it's all overkill for simple session tracking. Another problem with user authorization is that a user cannot simultaneously maintain more than one session at the same site.

Hidden Form Fields:

One way to support anonymous session tracking is to use hidden form fields. As the name implies, these are fields added to an HTML form that are not displayed in the client's browser, they are sent back to the server when the form that contains them is submitted.

In a sense, hidden form fields define constant variables for a form. For a servlet receiving a submitted form, there is no difference between a hidden field and a visible field.

As more information is associated with a client's session. It can become burdensome to pass it all using hidden form fields. In these situations, it's possible to pass on just a unique session ID that identifies a particular client session.

That session ID can be associated with complete information about its session that is stored on the server.

The advantage of hidden form fields is their ubiquity and support for anonymity. Hidden fields are supported in all the popular browsers, they demand no special server requirements, and they can be used with clients that haven't registered or logged in.

The major disadvantage with this technique, however, is that it works only for a sequence of dynamically generated forms, the technique breaks down immediately with static documents, emailed documents, bookmarked documents and browser shutdowns.

URL Rewriting:

URL rewriting is another way to support anonymous session tracking. With URL rewriting every local URL the user might click on is dynamically modified. Or rewritten, to include extra information. The extra information can be in the form of extra path information, added parameters, or some custom, server-specific URL change. Due to the limited space available in rewriting a URL, the extra information is usually limited to a unique session.

Each rewriting technique has its own advantage and disadvantage

Using extra path information works on all servers, and it works as a target for forms that use both the Get and Post methods. It does not work well if the servlet has to use the extra path information as true path information

The advantages and disadvantages of URL. Rewriting closely match those of hidden form fields, The major difference is that URL rewriting works for all dynamically created documents, such as the Help servlet, not just forms. With the right server support, custom URL rewriting can even work for static documents.

Persistent Cookies:

A fourth technique to perform session tracking involves persistent cookies. A cookie is a bit of information. Sent by a web server to a browser that can later be read back from that browser. When a browser receives a cookie, it saves the cookie and there after sending the cookie back to the server each time it accesses a page on that server, subject to certain rules. Because a cookie's value can uniquely identify a client, cookies are often used for session tracking.

Persistent cookies offer an elegant, efficient, easy way to implement session tracking. Cookies provide as automatic an introduction for each request, as we could hope for. For each request, a cookie can automatically provide a client's session ID or perhaps a list of clients' performance. The ability to customize cookies gives them extra power and versatility.

The biggest problem with cookies is that browsers don't always accept cookies sometimes this is because the browser doesn't support cookies. More often it's because the browser doesn't support cookies. More often it's because the user has specifically configured the browser to refuse cookies.

The power of servlets:

The power of servlets is nothing but the advantages of servlets over other approaches, which include portability, power, efficiency, endurance, safety, elegance, integration, extensibility and flexibility.

Portability:

As servlets are written in Java and conform to a well defined and widely accepted API. They are highly portable across operating systems and across server implementation

We can develop a servlet on a Windows NT machine running the Java web server and later deploy it effortlessly on a high-end UNIX server running Apache. With servlets we can really “write once, serve everywhere”

Servlet portability is not the stumbling block it so often is with applets, for two reasons

First, Servlet portability is not mandatory, i.e. servlets have to work only on server machines that we are using for development and deployment

Second, servlets avoid the most error-prone and inconstancy implemented portions of the Java languages.

Power:

Servlets can harness the full power of the core Java. API's: such as Networking and Url access, multithreading, image manipulation, data compression, database connectivity, internationalization, remote method invocation (RMI) CORBA connectivity, and object serialization, among others.

Efficiency and Endurance:

Servlet invocation is highly efficient, Once a servlet is loaded it generally remains in the server's memory as a single object instance, Thereafter the server invokes the servlet to handle a request using a simple, light weighted method invocation. Unlike the CGI, there's no process to spawn or interpreter to invoke, so the servlet can begin handling the request almost immediately, Multiple, concurrent requests is handled the request almost immediately. Multiple, concurrent requests are handled by separate threads, so servlets are highly scalable.

Servlets in general are enduring objects. Because a servlet stays in the server's memory as a single object instance. It automatically maintains its state and can hold onto external resources, such as database connections.

Safety:

Servlets support safe programming practices on a number of levels. As they are written in Java, servlets inherit stronger type safety of the Java language. In addition the servlet API is

implemented to be type safe. Java's automatic garbage collection and lack of pointers mean that servlets are generally safe from memory management problems like dangling pointers invalid pointer references and memory leaks.

Servlets can handle errors safely, due to Java exception – handling mechanism. If a servlet divides by zero or performs some illegal operations, it throws an exception that can be safely caught and handled by the server.

A server can further protect itself from servlets through the use of Java security manager. A server can execute its servlets under the watch of a strict security manager.

Elegance:

The elegance of the servlet code is striking. Servlet code is clean, object oriented modular and amazingly simple one reason for this simplicity is the served API itself. This includes methods and classes to handle many of the routine chores of servlet development. Even advanced to operations like cookie handling and session tracking are abstracted into convenient classes.

Integration:

Servlets are tightly integrated with the server. This integration allows a servlet to cooperate with the server in two ways. For e.g.: a servlet can use the server to translate file paths, perform logging, check authorization, perform MIME type mapping and in some cases even add users to the server's user database.

Extensibility and Flexibility:

The servlet API is designed to be easily extensible. As it stands today the API includes classes that are optimized for HTTP servlets. But later it can be extended and optimized for another type of servlets. It is also possible that its support for HTTP servlets could be further enhanced.

Servlets are also quite flexible; Sun also introduced Java server pages. Which offer a way to write snippets of servlet code directly within a static HTML page using syntax similar to Microsoft's Active server pages (ASP)

5.3 JDBC

What is JDBC?

Any relational database. One can write a single program using the JDBC API, and the JDBC is a Java API for executing SQL statements. (As a point of interest, JDBC are trademarks names and is not an acronym; nevertheless, Jdbc is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the Java Programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API.

Using JDBC, it is easy to send SQL statements to virtually any program will be able to send SQL statements to the appropriate database. The combination of Java and JDBC lets a programmer write it once and run it anywhere.

What Does JDBC Do?

Simply put, JDBC makes it possible to do three things

- Establish a connection with a database
- Send SQL statements
- Process the results
- JDBC Driver Types
- The JDBC drivers that we are aware of this time fit into one of four categories
- JDBC-ODBC Bridge plus ODBC driver
- Native-API party-Java driver
- JDBC-Net pure Java driver
- Native-protocol pure Java driver

An individual database system is accessed via a specific JDBC driver that implements the `java.sql.Driver` interface. Drivers exist for nearly all popular RDBMS systems, though few are available for free. Sun bundles a free JDBC-ODBC bridge driver with the JDK to allow access to a standard ODBC, data sources, such as a Microsoft Access database, Sun advises against using the bridge driver for anything other than development and very limited development.

JDBC drivers are available for most database platforms, from a number of vendors and in a number of different flavors. There are four driver categories

Type 01-JDBC-ODBC Bridge Driver

Type 01 drivers use a bridge technology to connect a Java client to an ODBC database service. Sun's JDBC-ODBC bridge is the most common type 01 driver. These drivers implemented using native code.

Type 02-Native-API party-Java Driver

Type 02 drivers wrap a thin layer of Java around a database-specific native code libraries for Oracle databases, the native code libraries might be based on the OCI(Oracle call Interface) libraries, which were originally designed for c/c++ programmers, Because type-02 drivers are implemented using native code. In some cases, they have better performance than their all-Java counterparts. They add an element of risk, however, because a defect in a driver's native code section can crash the entire server

Type 03-Net-Protocol All-Java Driver

Type 03 drivers communicate via a generic network protocol to a piece of custom middleware. The middleware component might use any type of driver to provide the actual database access. These drivers are all Java, which makes them useful for applet deployment and safe for servlet deployment

Type-04-native-protocol All-Java Driver

Type o4 drivers are the most direct of the lot. Written entirely in Java, Type 04 drivers understand database-specific networking. Protocols and can access the database directly without any additional software

JDBC-ODBC Bridge

If possible use a Pure Java JDBC driver instead of the Bridge and an ODBC driver. This completely eliminates the client configuration required by ODBC.It also eliminates the potential

that the Java VM could be corrupted by an error in the native code brought in by the Bridge (that is, the Bridge native library, the ODBC driver manager library, library, the ODBC driver library, and the database client library).

WHAT IS The JDBC-ODBE Bridge?

The JDBC-ODBC Bridge is a Jdbc driver, which implements JDBC operations by translating them into ODBC operations. To ODBC it appears as a normal application program. The Bridge is implemented as the sun.jdbc.odbc Java package and contains a native library used to access ODBC. The Bridge is a joint development of Intersolv and Java Soft

5.4 HTML, JAVASCRIPT

Hypertext Markup Language(HTML), the languages of the World Wide Web (WWW), allows users to produce web pages that included text, graphics and pointer to other web pages (Hyperlinks).

HTML is not a programming language, but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but Specialized to hypertext and adapted to the Web. The idea behind Hypertext one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop

HTML provides tags (special codes) to make the document look attractive.

HTML provides are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

Basic Html Tags:

<!-- -->	Specific Comments.
<A>.....	Creates Hypertext links.
.....	Creates hypertext links.
<Big>.....</Big>	Formats text in large-font
<Body>..... </Body>	contains all tags and text in the Html-document
<Center>.....</Center>	Creates Text
<DD>.....</DD>	Definition of a term.
<TABLE>..... </TABLE>	creates a table
<Td>.....</Td>	indicates table data in a table.
<Tr>.....</Tr>	designates a table row
<Th>..... </Th>	creates a heading in a table.

ADVANTAGE

- A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
- HTML is platform independent
- HTML tags are not case-sensitive.

5.5 JAVA SCRIPT

The Java Script Language

JavaScript is a compact, object-based scripting language for developing client and server internet applications. Netscape Navigator 2.0 interprets JavaScript statements embedded directly in an HTML page. And Livewire enables you to create server-based applications similar to common gateway interface (CGI) programs.

In a client application for Navigator, JavaScript statements embedded in an HTML Page can recognize and respond to user events such as mouse clicks form input, and page navigation.

For example, you can write a JavaScript function to verify that users enter valid information into a form requesting a telephone number or zip code. Without any network transmission, an Html page with embedded Java Script can interpret the entered text and alert the user with a message dialog if the input is invalid or you can use JavaScript to perform an action (such as play an audio file, execute an applet, or communicate with a plug-in) in response to the user opening or exiting a page.

CHAPTER-6

SYSTEM DESIGN

6.1 INTRODUCTION

In early stage people are aware of solar energy by commercial people are not affordable due to high cost of price. In upcoming world the solar panel types are classified into many segments, price is increase or decrease according to panel types. On that stage also the people have no knowledge about what panel is used for their purpose and also the waste their money. By this situation we want to the change people mind to give awareness about importance of energy through this application. By this application the people get know about the panel types and the importance also their features, so people can easily understand it. By some of the people can check with cost wise, this feature also available in this application. In this feature can be done with concept of feature selection method, in this method is used to give the relevant information to the user. In energy preserving scheme selection approach to predict the material properties through machine learning and computational methods. In this application, we apply algorithms for panel prediction through user data's; it is a crucial step of data driven machine learning approach where the panels and the related information on the datasets. When the user give the specific data as input and it process to get relevant information as output.

MODULES:

1. Panel Data Upload
2. Model Training
3. Customer or User
4. Staff verification
5. Admin

Panel data upload

In this module, the Panels related data such as types and other properties of a panels information will be uploaded. After that the training dataset will be collected, contains the overall panel information which is occurred in the past years. Then it will be preprocessed according to the scenario and it will move on to the next phase.

Model Training

In this module the model will be trained according to the given input datasets and other properties of the panel. Finally the model which is in the state to predict which the exact outflow of the given dam is. In some critical circumstances the model will predict and suggest information of panel according to their budget.

Customer or User

In this module the customer has to register the basic details once they register they show the pop up window register successfully, then the login page to login their account in this the user page will display in this the types of panels and their product description has give you can click each of the panel type will show the panel description and their relevant details. Then the next step use the get quote method click then show the land area budget(amount) we can enter the detail then the algorithm will take the land area and budget amount has a basic data or information and the fetches the relevant information about the panels according to the land area and amount is given by the user it will show the panel type according to budget is quoted. In this they show the panel types and their relevant details, once the buyer will choose their product according to their purpose, once they click buy and it will show the additional information page (i.e. address, zip, pincode, phone no) then the pop up the window your order places successfully, it will to user home you check your order list in the it show your order. Once the payment status comes from the admin side the customer can done the payment in the payments page, once the payment was successful then status will to success.

Staff Verification

In this module the staff has already register and approved by the admin side, and then the login they redirect to the home interface, in that we click the status then it will show the

customer order information then the staff go the specific location for the manual verification process then you give click the verification then it show the customer information and click the not verified then redirect to the verification in this they show the customer information along with uploaded image the cross verify manually then it will ok mean then the staff will approve then it will updated to the both admin and buyer.

Admin

In this module admin is the core part is to maintain the data of the product details they upload the product details then work of the admin is to check the customer order status and then staff manual verification status once the staff check manual verification process then they give approve means then the admin will check the availability of the stock in the customer product once the product is available then it proceed status to the customer for the payment process.

Database Screen Short:

Basic Options Indexes (0) Foreign keys (0) Check constraints (0) Partitions CREATE code

Name: image

Comment:

Columns: + Add - Remove Up Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment
1	uid	BIGINT	20	☐	☐	☐	'0'	
2	aadh	VARCHAR	100	☐	☒	☐	NULL	
3	land	VARCHAR	100	☐	☒	☐	NULL	

Help Discard Save

BasicOptionsIndexes (0)Foreign keys (0)Check constraints (0)PartitionsCREATE code

Name:reg

Comment:

Columns: Add Remove Up Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment
1	uid	BIGINT	20	☐	☐	☐	'0'	
2	FullName	VARCHAR	75	☐	☒	☐	NULL	
3	Email	VARCHAR	75	☐	☒	☐	NULL	
4	Phoneno	VARCHAR	50	☐	☒	☐	NULL	
5	password	VARCHAR	75	☐	☒	☐	NULL	

Basic Options Indexes (0) Foreign keys (0) Check constraints (0) Partitions CREATE code

Name: quote

Comment:

Columns: Add Remove Up Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment
1	uid	BIGINT	20	☐	☒	☐	NULL	
2	pid	VARCHAR	50	☐	☒	☐	NULL	

Help Discard Save

Basic Options Indexes (0) Foreign keys (0) Check constraints (0) Partitions CREATE code

Name: personreg

Comment:

Columns: Add Remove Up Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment
1	uid	BIGINT	20	☐	☐	☐	'0'	
2	product	VARCHAR	80	☐	☒	☐	NULL	
3	landarea	VARCHAR	50	☐	☒	☐	NULL	
4	name	VARCHAR	80	☐	☒	☐	NULL	
5	address	VARCHAR	80	☐	☒	☐	NULL	
6	district	VARCHAR	80	☐	☒	☐	NULL	
7	state	VARCHAR	80	☐	☒	☐	NULL	
8	pincode	VARCHAR	80	☐	☒	☐	NULL	
9	phoneno	VARCHAR	80	☐	☒	☐	NULL	
10	Date	VARCHAR	50	☐	☒	☐	NULL	
11	aadh	VARCHAR	50	☐	☒	☐	NULL	
12	land	VARCHAR	50	☐	☒	☐	NULL	
13	status	VARCHAR	50	☐	☒	☐	NULL	
14	amt	VARCHAR	50	☐	☒	☐	NULL	

Help Discard Save

Basic

Options

Indexes (0)

Foreign keys (0)

Check constraints (0)

Partitions

CREATE code

Name:

panels

Comment:

Columns:

+

 Add

×

 Remove

▲

 Up

▼

 Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment
1	pid	VARCHAR	50	☐	☒	☐	NULL	
2	productname	VARCHAR	500	☐	☒	☐	NULL	
3	kw	VARCHAR	500	☐	☒	☐	NULL	
4	key	VARCHAR	500	☐	☒	☐	NULL	
5	battery	VARCHAR	500	☐	☒	☐	NULL	
6	inveter	VARCHAR	500	☐	☒	☐	NULL	
7	mount	VARCHAR	500	☐	☒	☐	NULL	
8	area	VARCHAR	500	☐	☒	☐	NULL	
9	warranty	VARCHAR	500	☐	☒	☐	NULL	
10	labour	VARCHAR	500	☐	☒	☐	NULL	
11	amt	VARCHAR	500	☐	☒	☐	NULL	

<

>

Help

Discard

Save

Basic

Options

Indexes (0)

Foreign keys (0)

Check constraints (0)

Partitions

CREATE code

Name:

panelbued

Comment:

Columns:

+

 Add

×

 Remove

▲

 Up

▼

 Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment
1	uid	BIGINT	20	☐	☒	☐	NULL	
2	pid	VARCHAR	50	☐	☒	☐	NULL	
3	productname	VARCHAR	100	☐	☒	☐	NULL	
4	kw	VARCHAR	100	☐	☒	☐	NULL	
5	battery	VARCHAR	100	☐	☒	☐	NULL	
6	inveter	VARCHAR	100	☐	☒	☐	NULL	
7	mount	VARCHAR	100	☐	☒	☐	NULL	
8	area	VARCHAR	80	☐	☒	☐	NULL	
9	warranty	VARCHAR	50	☐	☒	☐	NULL	
10	labour	VARCHAR	50	☐	☒	☐	NULL	
11	amt	VARCHAR	50	☐	☒	☐	NULL	

<

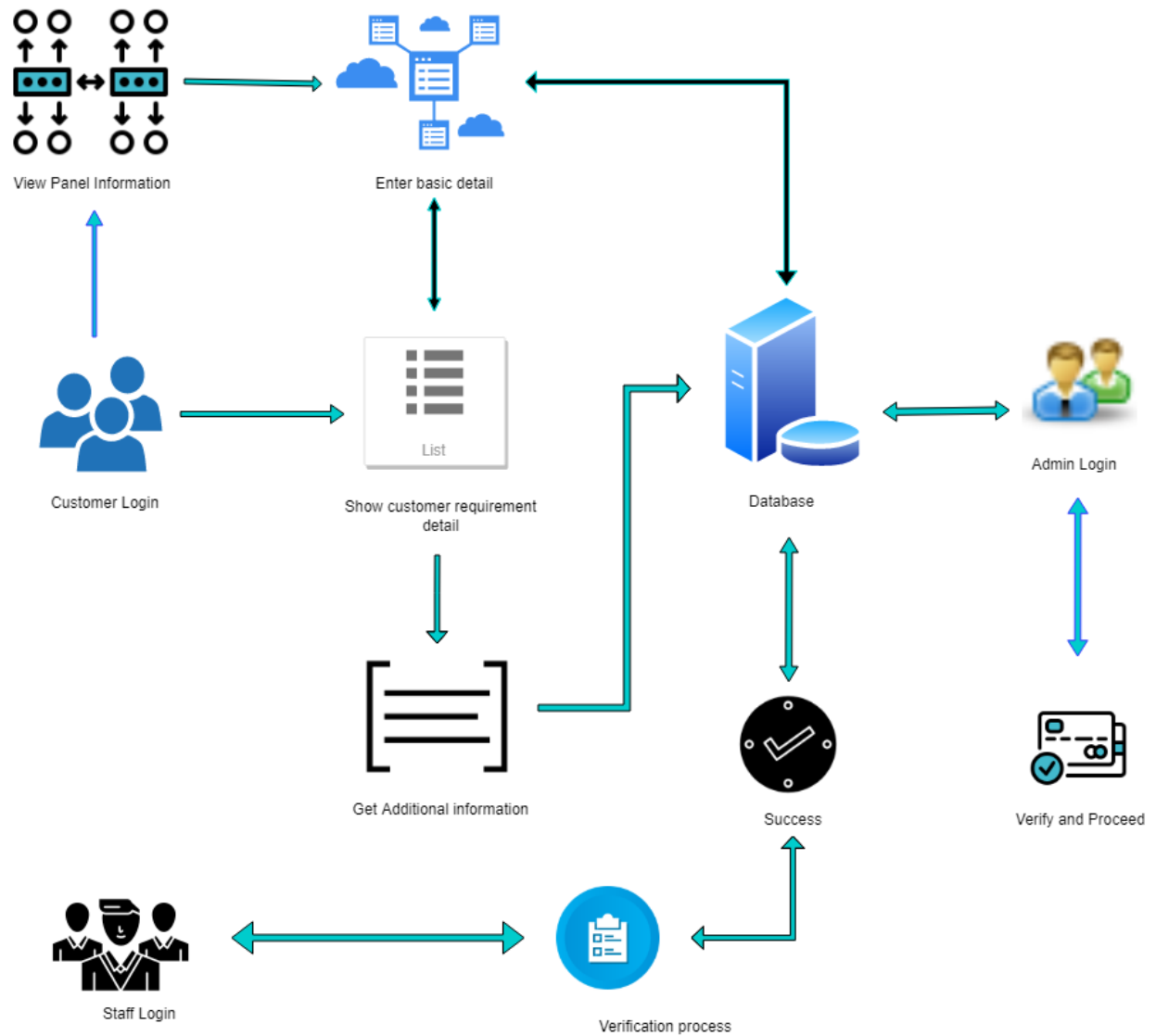
>

Help

Discard

Save

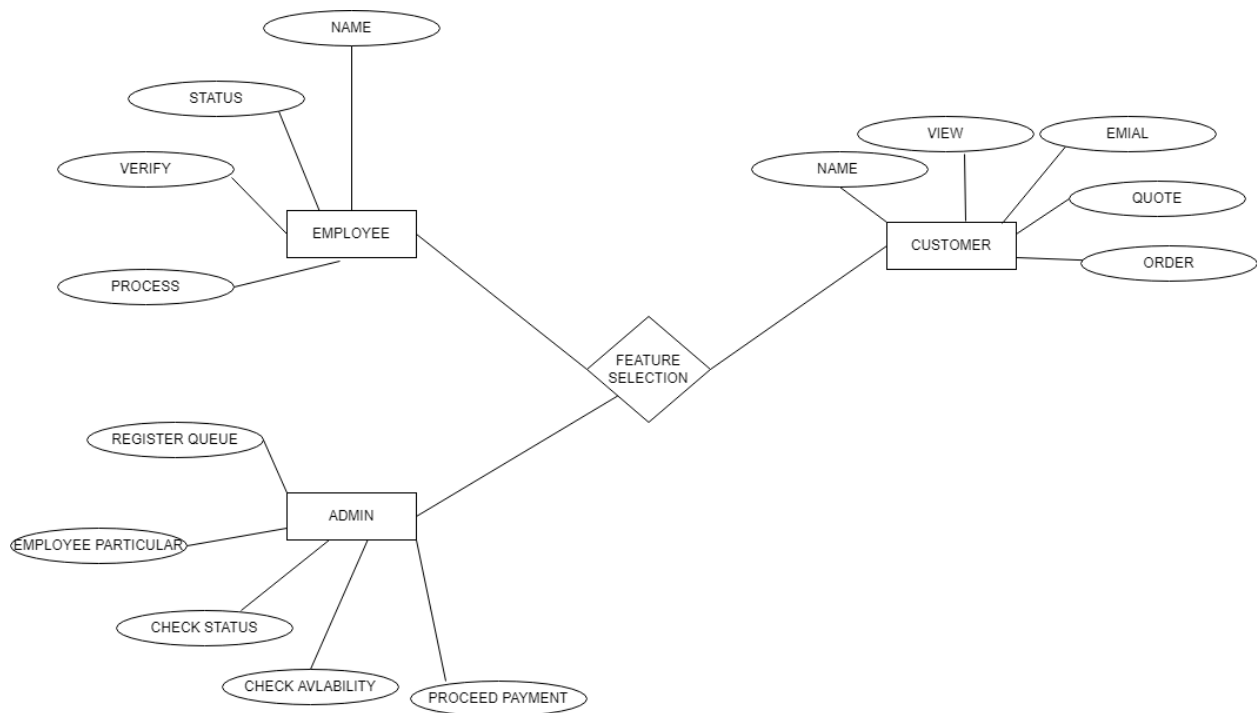
6.3 System Architected:



6.4 E – R DIAGRAMS

- The relation upon the system is structured through a conceptual ER-Diagram, which not only specifies the existing entities, but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.
- The Entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct, the data modeling activity the attributes of each data object noted, is the ERD can be described design a data object description.
- The set of primary components that are identified by the ERD are
 - Data object
 - Relationships
 - Attributes
 - Various types of indicators.

The primary purpose of the ERD is to represent data objects and their relationships.



6.5 FLOW DIAGRAMS

A data flow diagram is a graphical tool used to describe and analyze the movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output through processing, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. The process is further identified with a number that will be used for identification purpose. The development of DFD'S is done on several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The top-level diagram is often called context diagram. It consist a single process bit, which plays a vital role in studying the current system. The process in the context level diagram is exploded into another process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analysts to understand the process.

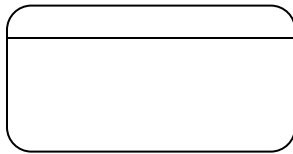
Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical form, this lead to the modular design.

A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programmed in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

6.6 DFD SYMBOLS

In the DFD, there are four symbols

1. A square defines a source (originating) or destination of system data
2. An arrow identifies data flow. It is the pipeline through which the information flows
3. A circle or a bubble represents a process that transforms the incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data



A process that transforms the data flow



Source or Destination of data



Data flow



Data Store

CONSTRUCTING A DFD:

Several rules of thumb are used in drawing DFD'S:

1. Process should be named and numbered for an easy reference. Each name should be representative of the process.

2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw the long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower level details, they are numbered.
4. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each word capitalized

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

Questionnaires should contain all the data elements that flow in and out. Missing interfaces redundancies and like is then accounted for often through interviews.

SAILENT FEATURES OF DFD'S

1. The DFD shows the flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.
2. The DFD does not indicate the time factor involved in any process, whether the dataflow take place daily, weekly, monthly or yearly.
3. The sequence of events is not brought out on the DFD.

TYPES OF DATA FLOW DIAGRAMS

1. Current Physical
2. Current Logical
3. New Logical
4. New Physical

CURRENT PHYSICAL

In Current Physical DFD process label includes the name of the people or their positions or the names of computer systems that might provide some of the overall system-processing label includes an identification of the technology used to process the data. Similarly, data flows and

data stores are often labelled with the names of the actual physical media on which data are stored such as file folders, computer files, business forms or computer tapes.

CURRENT LOGICAL:

The physical aspects in the system are removed as much as possible so that the current system is reduced to its essence to the data and the processes that transform them regardless of actual physical form.

NEW LOGICAL:

This is exactly like a current logical model if the user were completely happy with the user were completely happy with the functionality of the current system, but had problems with how it was implemented typically through the new logical model will differ from the current logical model while having additional functions, absolute function removal and inefficient flows recognized.

NEW PHYSICAL:

The new physical represents only the physical implementation of the new system.

RULES GOVERNING THE DFD'S

PROCESS

- 1) No process can have only outputs.
- 2) No process can have only inputs. If an object has only inputs than it must be a sink.
- 3) A process has a verb phrase label.

DATA STORE

- 1) Data cannot move directly from one data store to another data store, a process must move data.
- 2) Data cannot move directly from an outside source to a data store, a process, which receives, must move data from the source and place the data into the data store
- 3) A data store has a noun phrase label.

SOURCE OR SINK

The origin and /or destination of data

- 1) Data cannot move directly from a source to sink it must be moved from a process
- 2) A source and /or sink have a noun phrase label

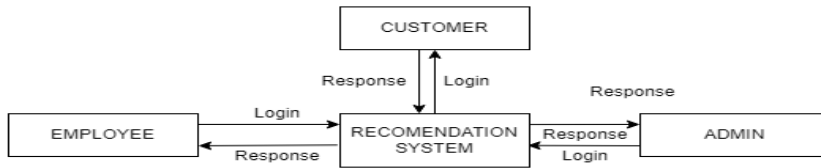
DATA FLOW

- 1) A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The latter is usually indicated, however by two separate arrows since these happen at different type.
- 2) A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location.
- 3) A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow produce some other data flow returns the original data in the beginning process.
- 4) A Data flow to a data store means update (delete or change).
- 5) A data Flow from a data store means retrieve or use.

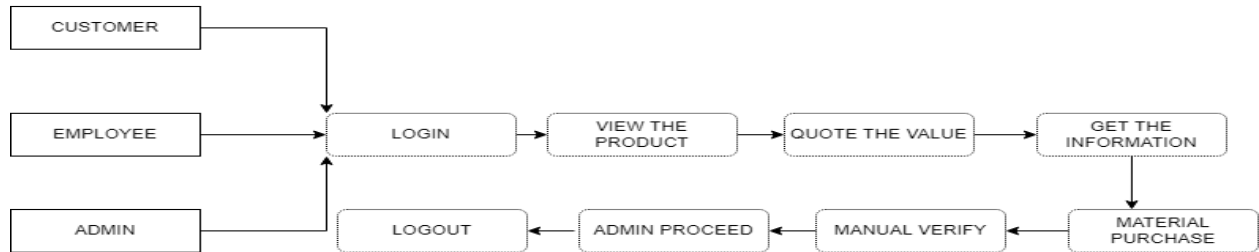
A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

DATA FLOW DIAGRAM

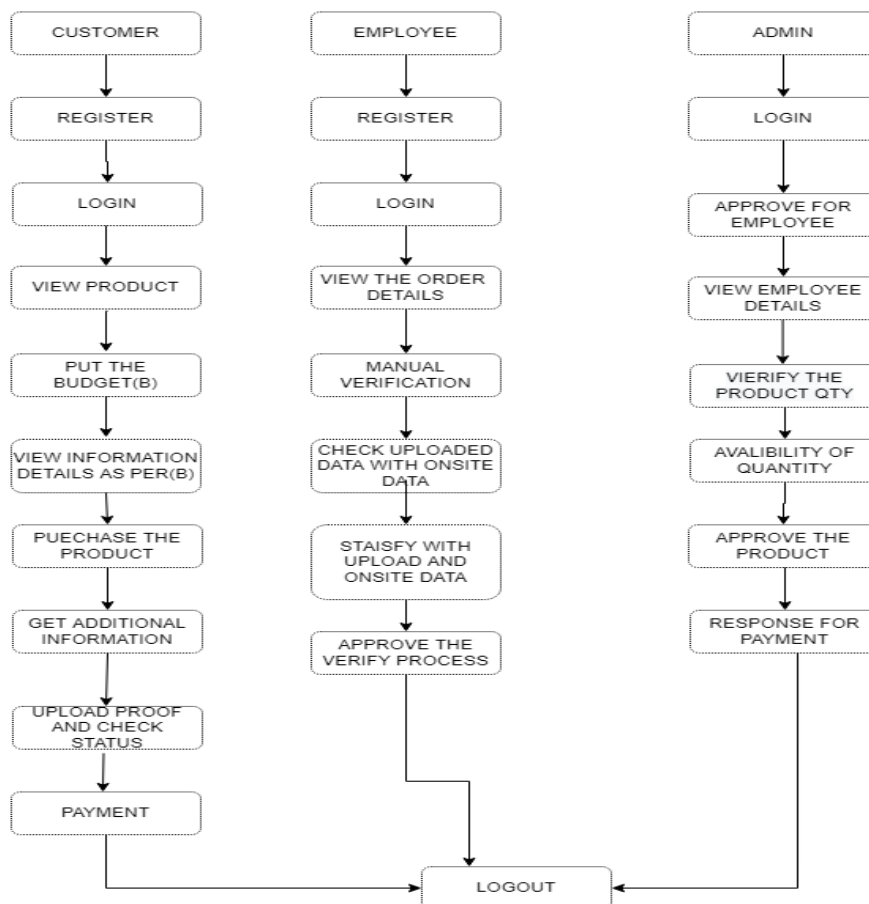
LEVEL 0



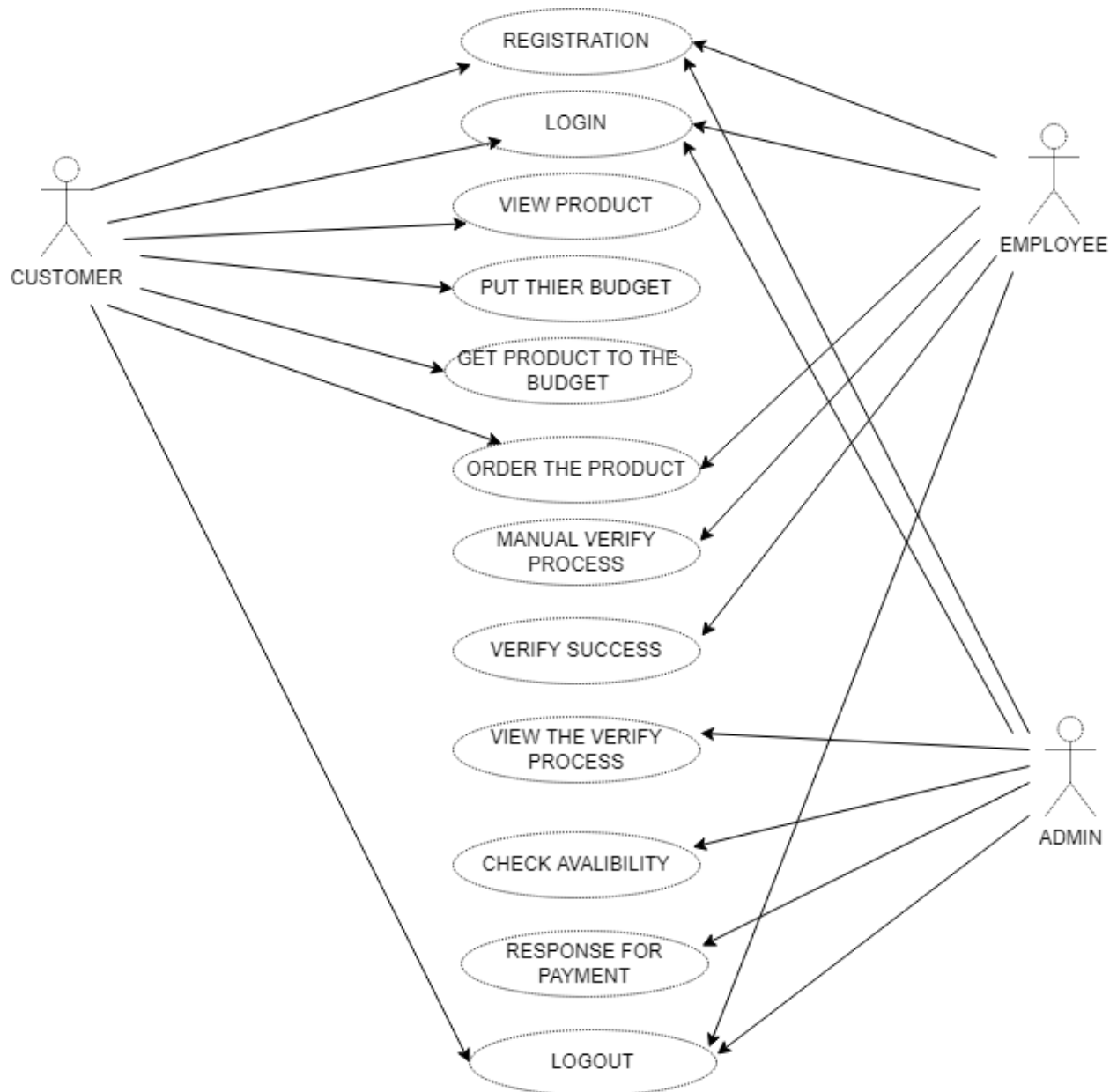
LEVEL 1



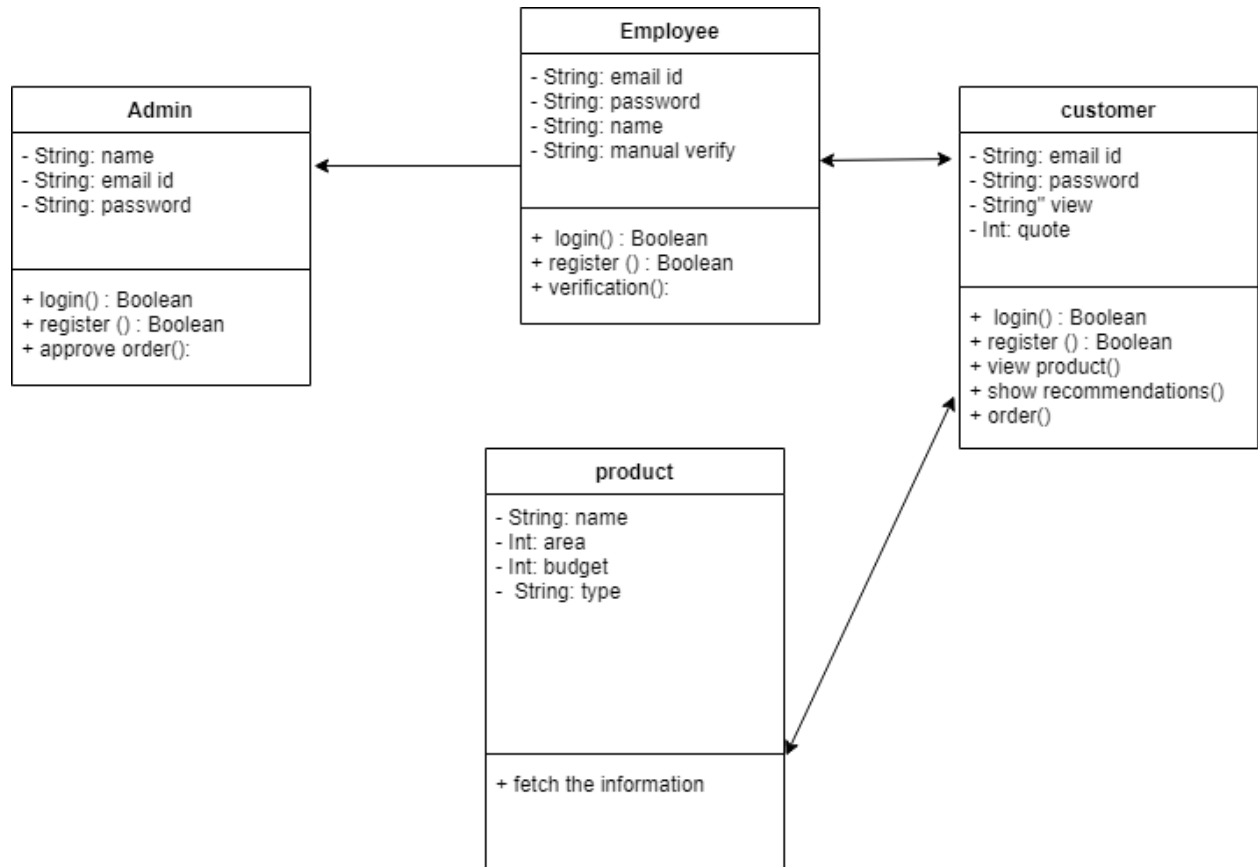
LEVEL 2



6.8 USE CASE DIAGRAM

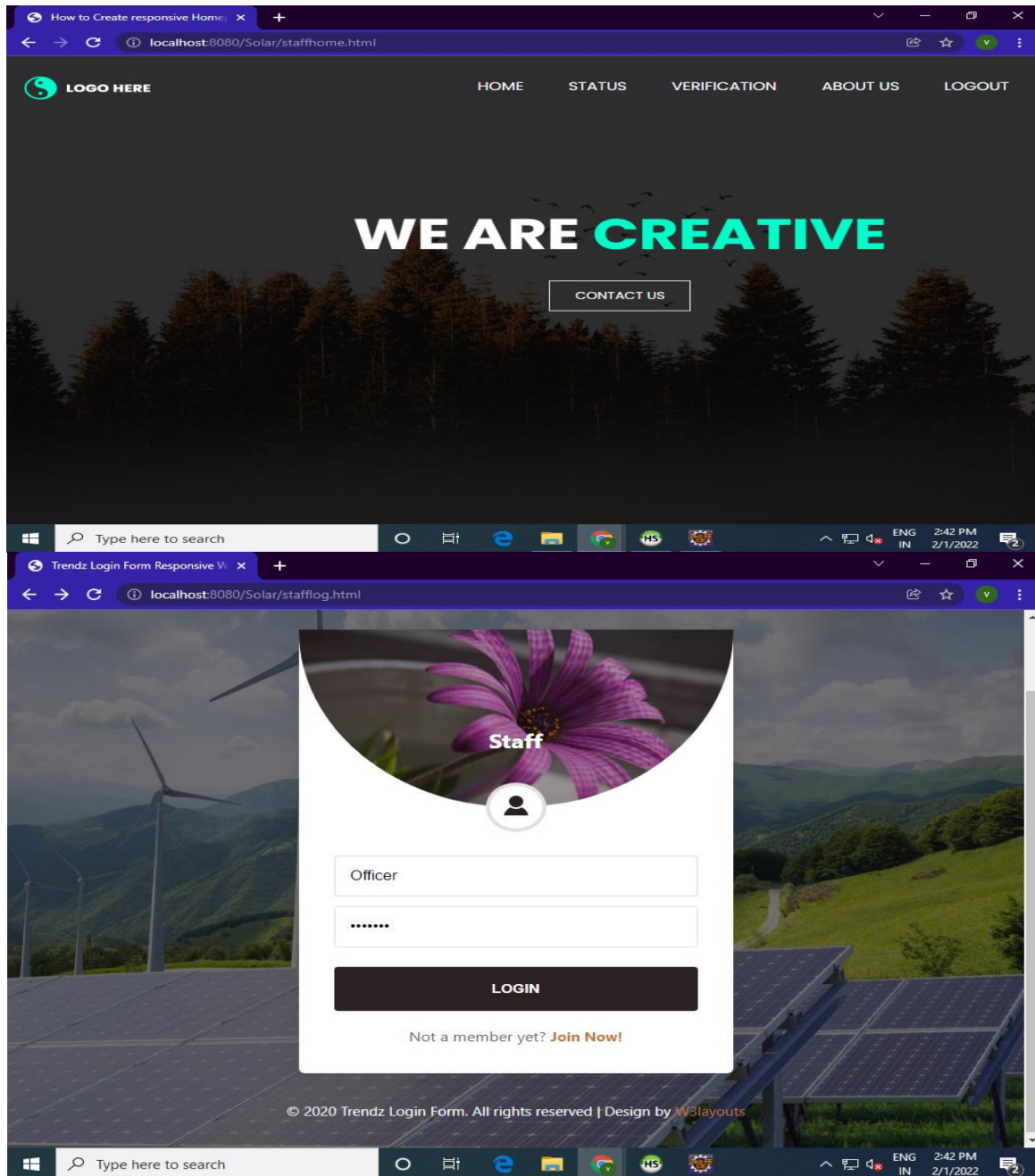


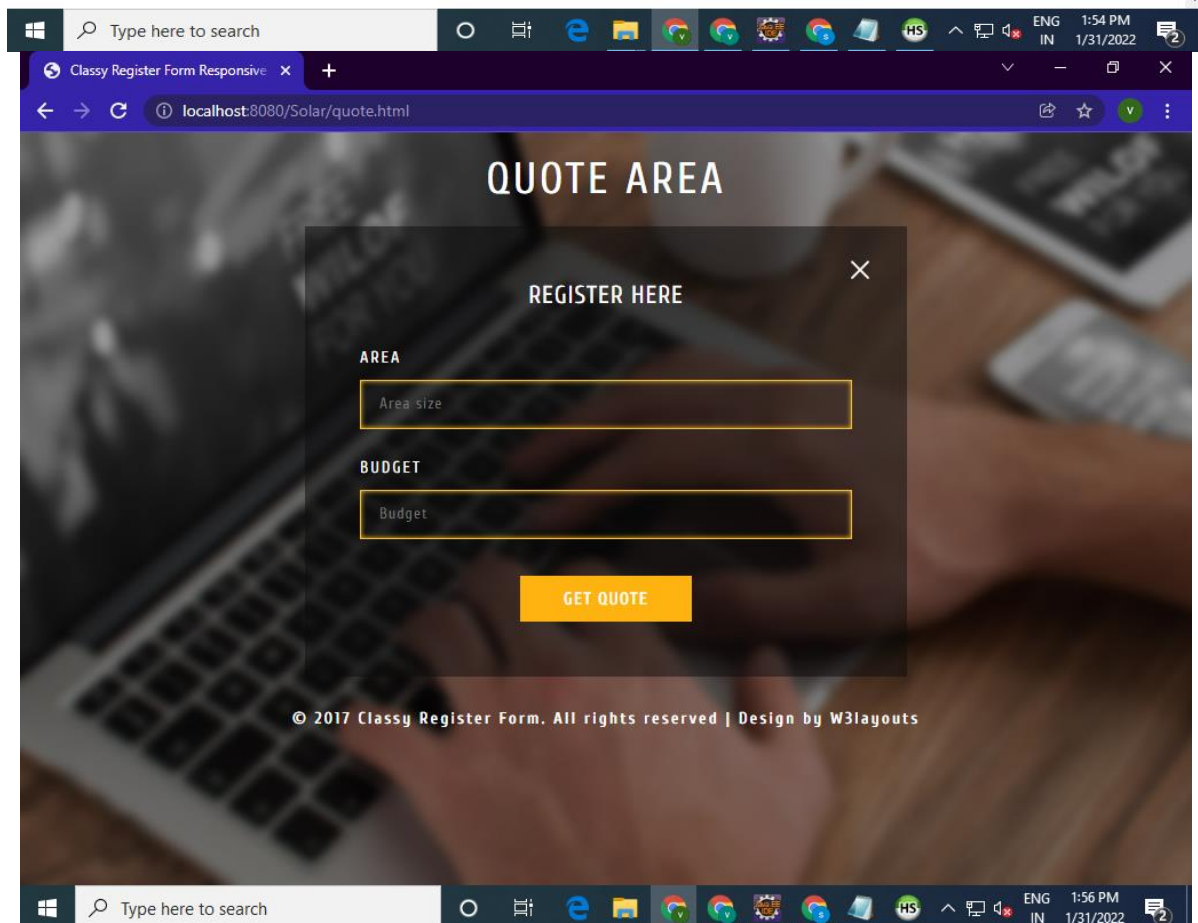
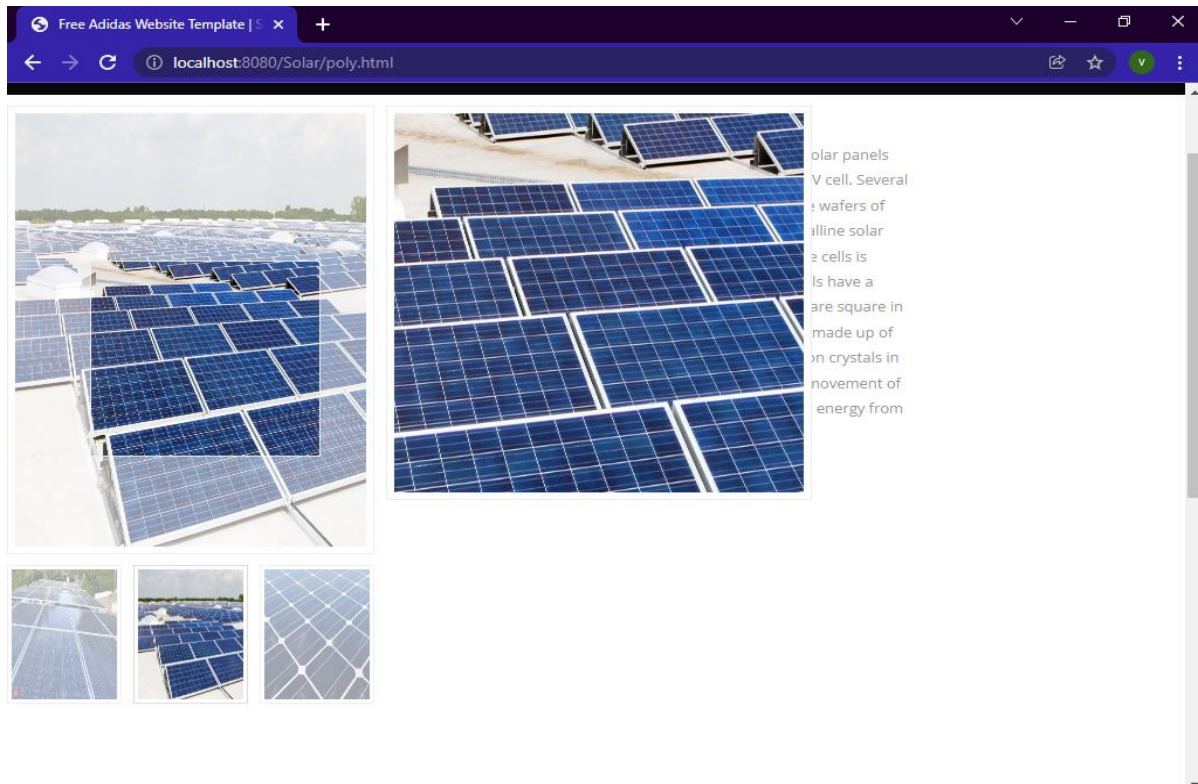
6.10 CLASS DIAGRAM



CHAPTER-7

OUTPUT SCREENS





Free Adidas Website Template | 5 x

localhost:8080/Solar/poly.html

SIGN UP & SAVESTORE LOCATORMY ACCOUNTCHECKOUT

HomeGet QuoteBuy ProductsApprovePaymentsLogout



POLY CRYSTALLINE PANEL

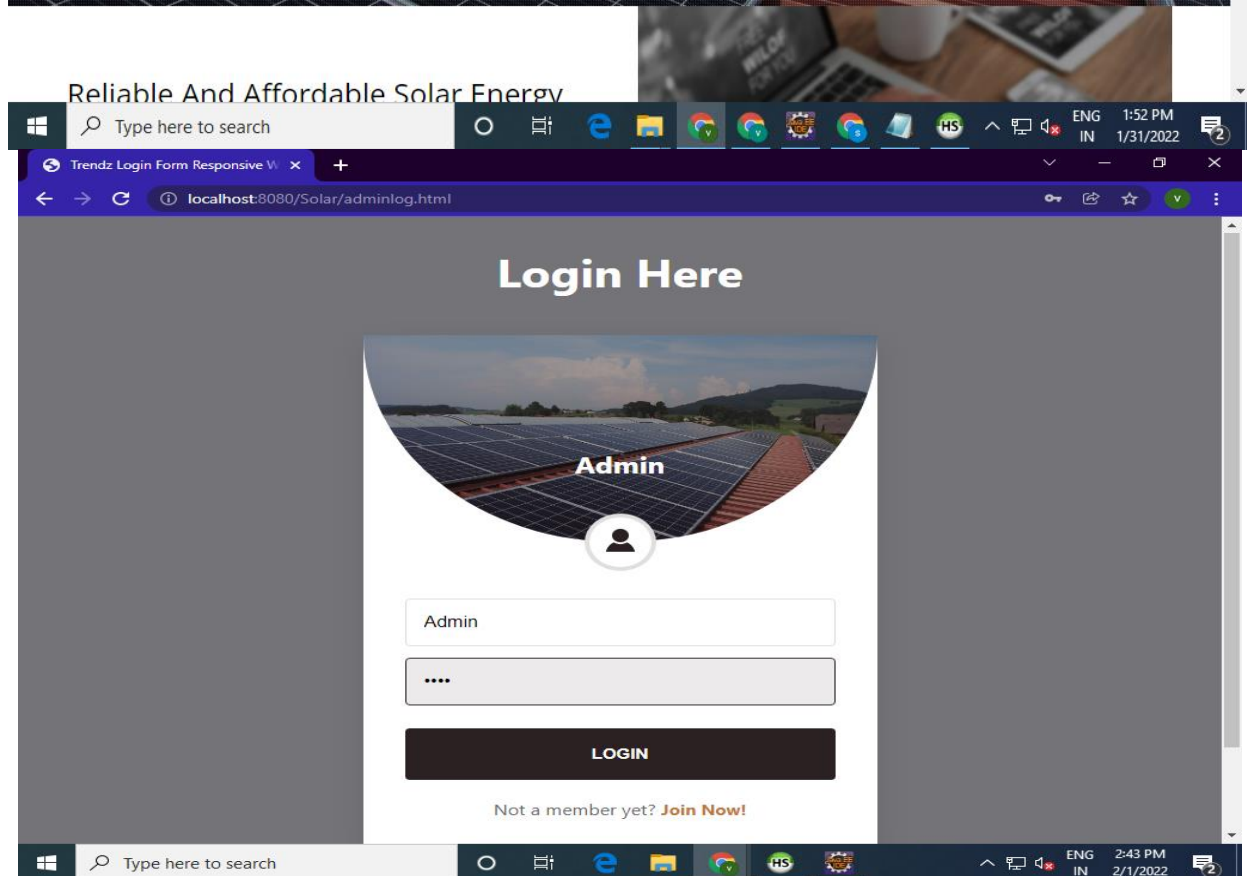
PolyCrystalline or MultiCrystalline solar panels are solar panels that consist of several crystals of silicon in a single PV cell. Several fragments of silicon are melted together to form the wafers of polycrystalline solar panels. In the case of polycrystalline solar panels, the vat of molten silicon used to produce the cells is allowed to cool on the panel itself. These solar panels have a surface that looks like a mosaic. These solar panels are square in shape and they have a shining blue hue as they are made up of several crystals of silicon. As there are multiple silicon crystals in each cell, so polycrystalline solar panels allow little movement of electrons inside the cells. These solar panels absorb energy from the sun and convert it into electricity.

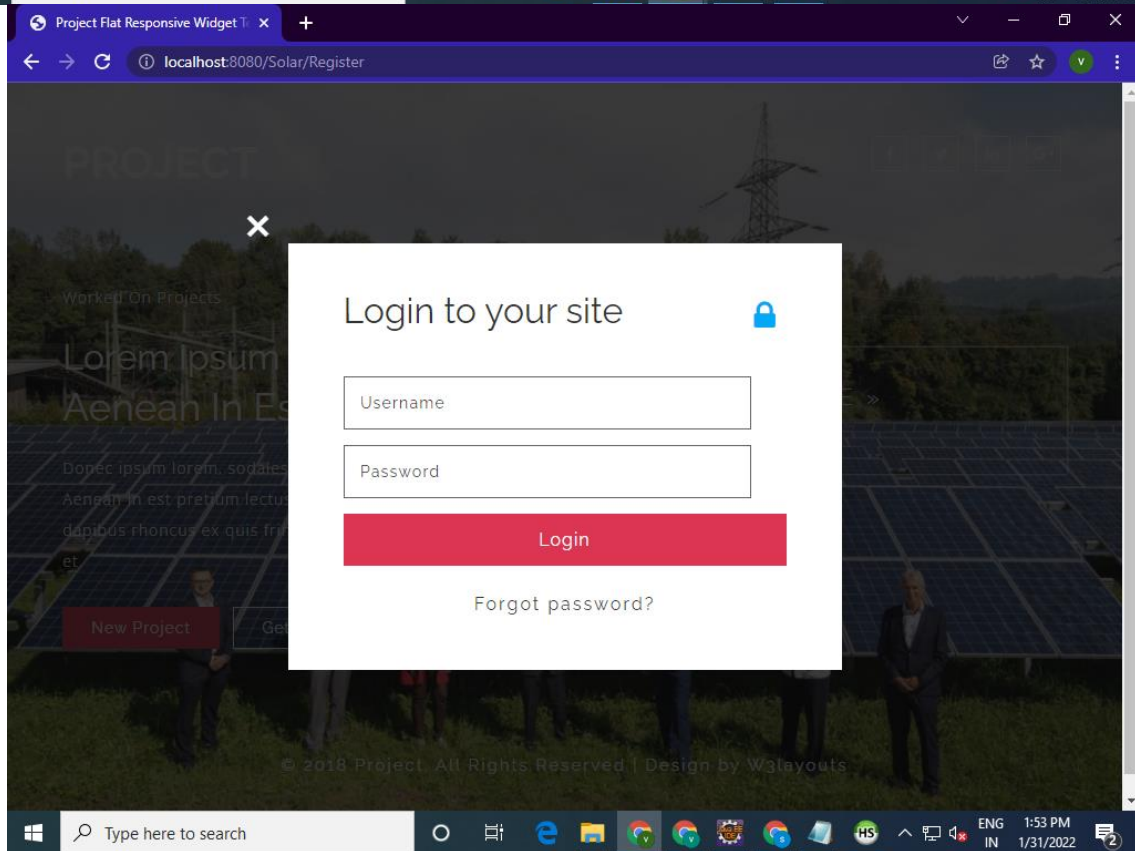
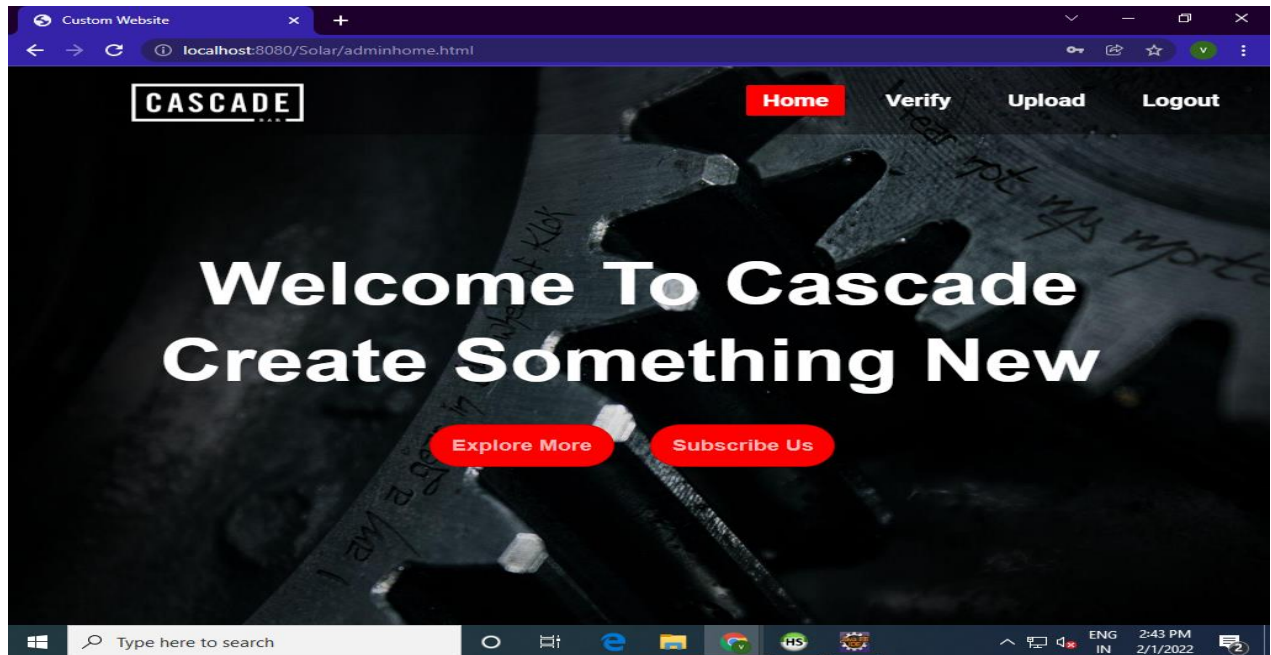


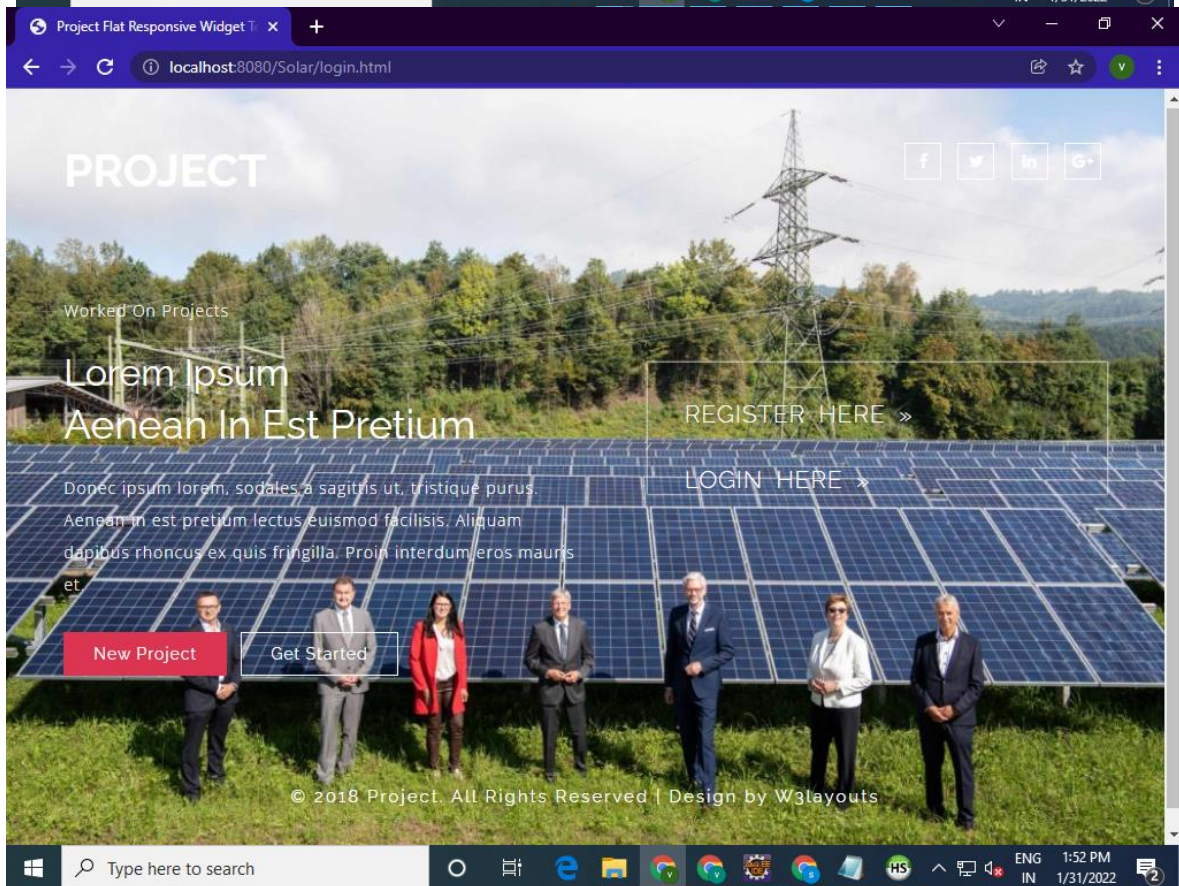
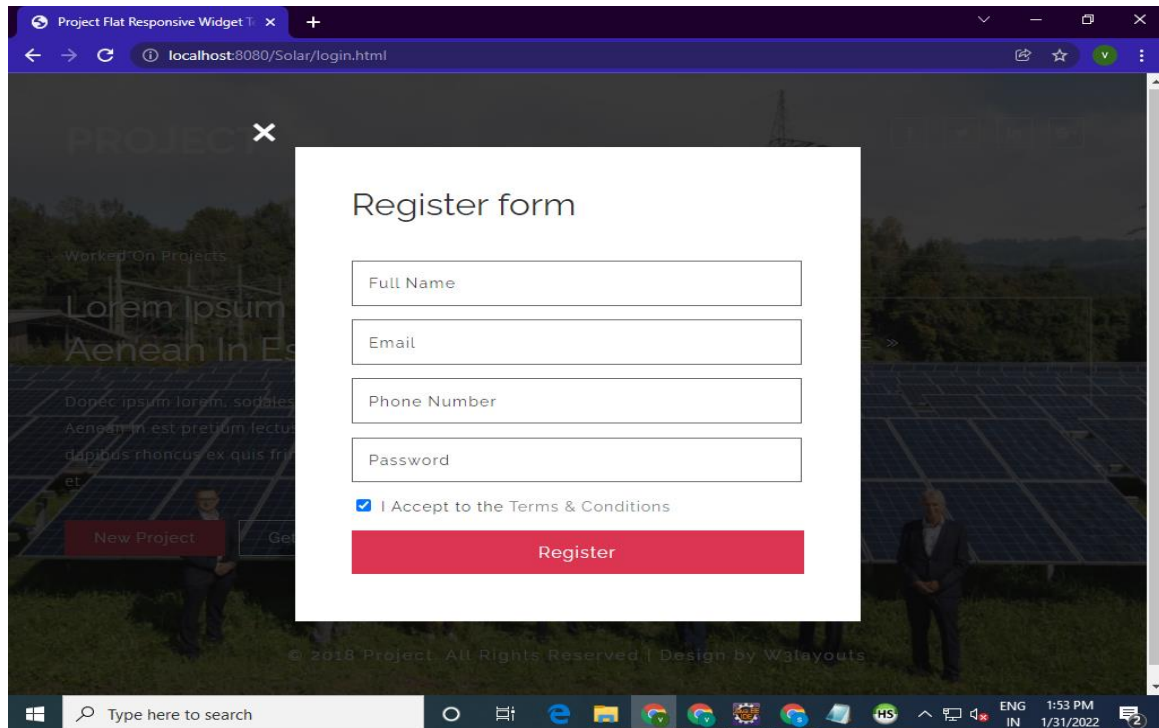
Type here to search

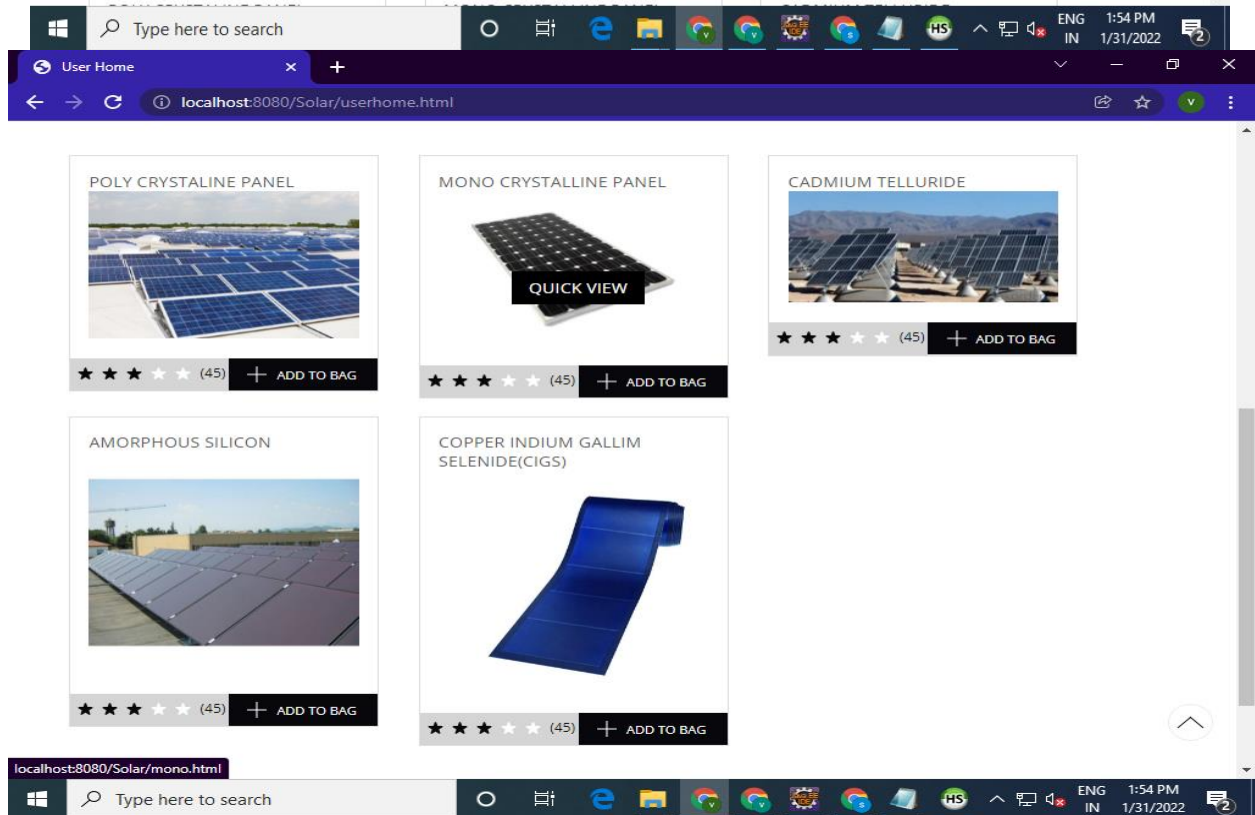
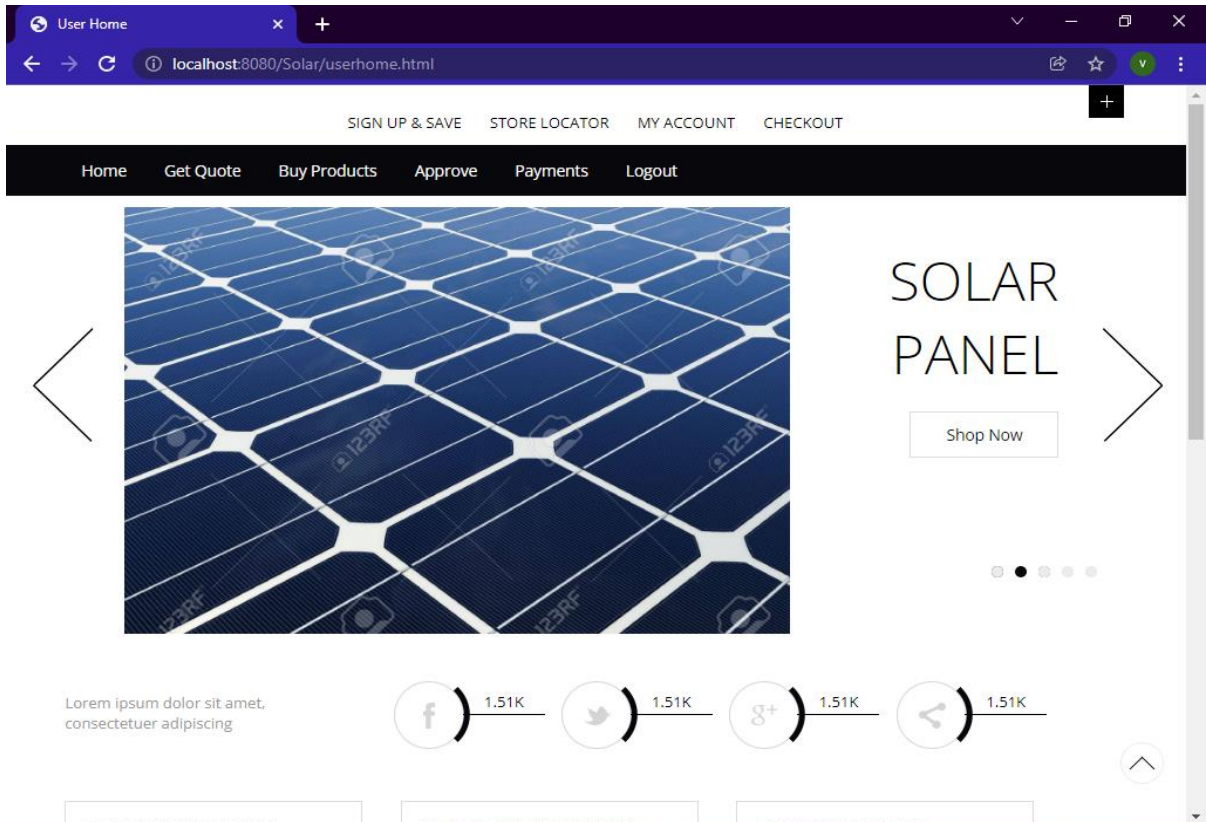


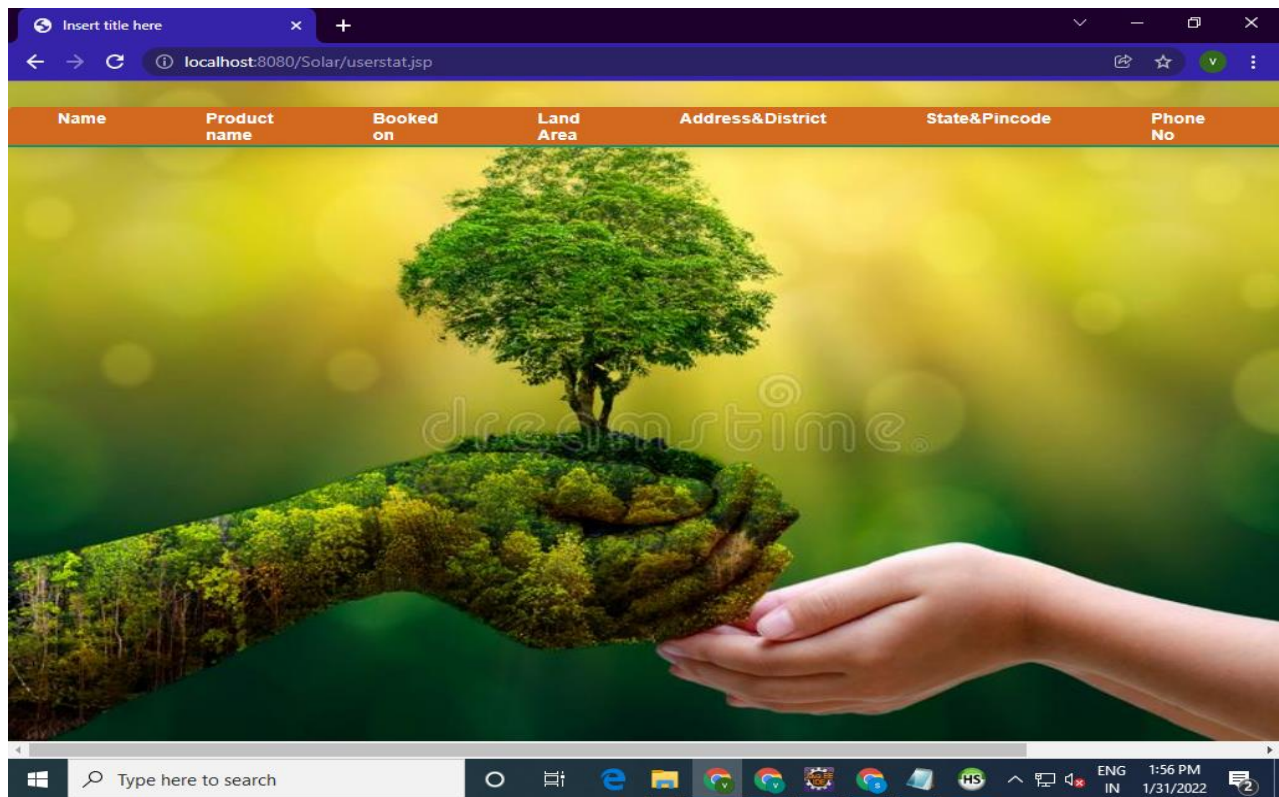
ENG1:55 PM1/31/2022











CHAPTER-8

CODINGS

RESOURCES:

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 * Servlet implementation class ADMINLOG
 */
@WebServlet("/admin")
public class admin extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public admin() {
        super();
        // TODO Auto-generated constructor stub
    }
}
```

```

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    handleRequest(request, response);
}

    public void handleRequest(HttpServletRequest req, HttpServletResponse resp) throws
IOException {

        HttpSession ses = req.getSession();
        resp.setContentType("text/html");

        // Post Parameters From The Request
        String param1 = req.getParameter("email");
        String param2 = req.getParameter("pass");

        if(param1 == null || param2 == null) {
            // The Request Parameters Were Not Present In The Query String. Do Something Or
            Exception Handling !!
        } else if ("".equals(param1) || "".equals(param2)) {
            // The Request Parameters Were Present In The Query String But Has No Value. Do
            Something Or Exception Handling !!
        } else {
            System.out.println("name?= " + param1 + ", password?= " + param2);

```

```
// Print The Response

PrintWriter out = resp.getWriter();


// Authentication Logic & Building The Html Response Code
if((param1.equalsIgnoreCase("Admin")) && (param2.equals("Admin"))) {
//ses.setAttribute("name", arg1);

    resp.sendRedirect("adminhome.html");
} else {
    resp.sendRedirect("adminlog.html");

}

out.write("</div></body></html>");
out.close();
}

}

import java.io.IOException;
import java.io.PrintWriter;


import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
```

```

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import utility.DB;

/**
 * Servlet implementation class datatrain
 */
@WebServlet("/datatrain")
public class datatrain extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        String csv = "E:/Solar/Datasets";
        String csvFilePath = csv+"//data.csv";
        String fname = request.getParameter("traindata");
        System.out.println(fname);
        PrintWriter out = response.getWriter();

        try {
            String qry = "LOAD DATA INFILE '"+csvFilePath+"' +
                "INTO TABLE panels FIELDS TERMINATED BY ',' +

```

```

        "OPTIONALLY ENCLOSED by \"\"";

        int i = DB.getconnection().prepareStatement(qry).executeUpdate();

        if(i>0) {
            out.print("<html><body><script>alert('Data Uploaded
Successfully');</script></body></html>");

            RequestDispatcher rs =
request.getRequestDispatcher("adminhome.html");

            rs.include(request, response);

        }else {

            response.sendRedirect("uploaddata.html");

        }

    }catch(Exception e) {

        e.printStackTrace();

    }

}

}

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

```

```

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;


import utility.DB;


/**
 * Servlet implementation class Login
 */
@WebServlet("/Login1")
public class Login1 extends HttpServlet {

    private static final long serialVersionUID = 1L;


    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out=response.getWriter();

        String n=request.getParameter("email");

        String p=request.getParameter("password");

        System.out.println(n+" "+p);

        try {

            String sel="select * from reg where Email=? and Password=?";

```

```

        PreparedStatement as = DB.getconnection().prepareStatement(sel);
        as.setString(1, n);
        as.setString(2, p);
        ResultSet rs = as.executeQuery();
        //HttpSession session = request.getSession();
        if (rs.next()) {
            //session.setAttribute("name", n);
            response.sendRedirect("userhome.html");
        } else {
            out.print("<head><body><script>alert('Sorry username or password
error')</script></body></html>");
            response.sendRedirect("login.html");
        }
        } catch (Exception e) {
            e.printStackTrace();
        }
        out.close();
    }
}

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

```

```
import javax.servlet.http.HttpSession;
```

```
/**
```

```
 * Servlet implementation class ADMINLOG
```

```
 */
```

```
@WebServlet("/logstaff")
```

```
public class logstaff extends HttpServlet {
```

```
private static final long serialVersionUID = 1L;
```

```
/**
```

```
 * @see HttpServlet#HttpServlet()
```

```
 */
```

```
public logstaff() {
```

```
    super();
```

```
    // TODO Auto-generated constructor stub
```

```
}
```

```
/**
```

```
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
```

```
 */
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException {
```

```
    handleRequest(request, response);
```

```
}
```

```
public void handleRequest(HttpServletRequest req, HttpServletResponse resp) throws
IOException {

    HttpSession ses = req.getSession();
    resp.setContentType("text/html");

    // Post Parameters From The Request
    String param1 = req.getParameter("email");
    String param2 = req.getParameter("password");

    if(param1 == null || param2 == null) {
        // The Request Parameters Were Not Present In The Query String. Do Something Or
        Exception Handling !!
    } else if ("".equals(param1) || "".equals(param2)) {
        // The Request Parameters Were Present In The Query String But Has No Value. Do
        Something Or Exception Handling !!
    } else {
        System.out.println("name?= " + param1 + ", password?= " + param2);

        // Print The Response
        PrintWriter out = resp.getWriter();

        // Authentication Logic & Building The Html Response Code
        if((param1.equalsIgnoreCase("Officer")) && (param2.equalsIgnoreCase("Officer"))) {
            //ses.setAttribute("name", arg1);
            resp.sendRedirect("staffhome.html");
```

```
        } else {  
            resp.sendRedirect("stafflog.html");  
  
        }  
        out.write("</div></body></html>");  
        out.close();  
    }  
  
}}
```

```
import java.io.IOException;  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import javax.servlet.http.HttpSession;
```

```
import utility.DB;
```

```
/**
```

```
 * Servlet implementation class paymentup
```

```
 */
```

```
@WebServlet("/paymentup")
```

```
public class paymentup extends HttpServlet {
```

```

private static final long serialVersionUID = 1L;

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

    HttpSession ses = request.getSession();

    int id17 = (Integer)ses.getAttribute("uid");

    String a15=request.getParameter("amont");

    String a13="PAYMENT SUCCESS ";

    try {

        String sql=("update personreg set status='"+ a13 +" where
uid='"+id17+"");

        int ps = DB.getconnection().prepareStatement(sql).executeUpdate();

        if(ps>0){

            response.sendRedirect("userhome.html");

        }

    } catch(Exception e){

        e.printStackTrace();

    }

}
}

```

```
import java.io.IOException;
```

```
import java.io.PrintWriter;
```

```
import java.sql.PreparedStatement;
```

```
import java.text.SimpleDateFormat;
```

```
import java.util.Random;
```

```
import javax.servlet.ServletException;
```

```
import javax.servlet.annotation.WebServlet;
```

```
import javax.servlet.http.HttpServlet;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
```

```
import javax.servlet.http.HttpSession;
```

```
import utility.DB;
```

```
/**
```

```
 * Servlet implementation class personreg
```

```
 */
```

```
@WebServlet("/personreg")
```

```
public class personreg extends HttpServlet {
```

```
    private static final long serialVersionUID = 1L;
```

```
    /**
```

```
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
```

```
     */
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
```

```
    response.setContentType("text/html");
```

```
    PrintWriter out = response.getWriter();
```

```
    String n=request.getParameter("product");
```

```
    String q=request.getParameter("landarea");
```

```
    String p=request.getParameter("name");
```

```
    String e=request.getParameter("add");
```

```
    String c=request.getParameter("dis");
```

```
    String z=request.getParameter("state");
```

```
    String f=request.getParameter("pin");
```

```
    String l=request.getParameter("pno");
```

```
    String a=request.getParameter("Amt");
```

```
    java.util.Date now = new java.util.Date();
```

```
    String date=now.toString();
```

```
    String DATE_FORMAT = "dd-MM-yyyy";
```

```
    SimpleDateFormat sdf = new SimpleDateFormat(DATE_FORMAT);
```

```
    String sdn = sdf.format(now) ;
```

```
    HttpSession ses = request.getSession();
```

```
    int s = (Integer)ses.getAttribute("uid");
```

```
    try{
```

```
        String sql ="insert into personreg values (?,?,?,?,?,?,?,?,?,?,?,?,?)" ;
```

```
PreparedStatement ps = DB.getconnection().prepareStatement(sql);
```

```
ps.setLong(1,s);  
ps.setString(2,n);  
ps.setString(3,q);  
ps.setString(4,p);  
ps.setString(5,e);  
ps.setString(6,c);  
ps.setString(7,z);  
ps.setString(8,f);  
ps.setString(9,l);  
ps.setString(10, sdn);  
ps.setString(11, "null");  
ps.setString(12, "null");  
ps.setString(13, "PENDING");  
ps.setString(14, a);
```

```
int i=ps.executeUpdate();  
if(i>0){  
    out.print("You are successfully registered...");  
    response.sendRedirect("adhar.jsp");  
}
```

```
} catch (Exception e2) {System.out.println(e2);} }
```

```
        out.close();
    }

}
```

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
```

```
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
```

```
import utility.DB;
```

```
/**
 * Servlet implementation class Process
 */
```

```

@WebServlet("/Process")

public class Process extends HttpServlet {

    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

        String area=request.getParameter("areas");

        String amt=request.getParameter("bud");

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        HttpSession ses = request.getSession();

        int id = (Integer)ses.getAttribute("uid");


        ResultSet rs1 = null;

        Statement st = null;

        ResultSet rs11=null;

        String
uid1=null,area1=null,amt1=null,pid=null,productname=null,kw=null,key=null,battery=null,invet
er=null,mount=null,area2=null,warranty=null,labour=null,amt2=null;


        try
        {

            String qry = "insert into quote values('"+id+"','"+area+"','"+amt+"')";

            int i = DB.getconnection().prepareStatement(qry).executeUpdate();

```

```
        if(i>0){

        }

    }

    catch(Exception e1)
    {
        e1.printStackTrace();
    }

    try{

        String qry1 = "SELECT * FROM quote WHERE uid='"+id+"'";
        rs1 = DB.getConnection().createStatement().executeQuery(qry1);
        if(rs1.next())
        {

            uid1 =rs1.getString("uid");
            area1=rs1.getString("area");
            amt1=rs11.getString("amount");
            System.out.println("guru");

        }
    }
    catch(Exception e1)
    {
        e1.printStackTrace();
    }
```

```

        if(amt.equalsIgnoreCase(amt1)||amt.equalsIgnoreCase("Any")){

            try{

                String qry2 = "SELECT * FROM panels WHERE

amt='"+amt1+"'";

                rs11 = DB.getconnection().createStatement().executeQuery(qry2);
                if(rs11.next()){
                    pid = rs11.getString("pid");
                    productname = rs11.getString("productname");
                    kw = rs11.getString("kw");
                    key = rs11.getString("key");
                    battery = rs11.getString("battery");
                    inveter = rs11.getString("inveter");
                    mount = rs11.getString("mount");
                    area2 = rs11.getString("area");
                    warranty = rs11.getString("warranty");
                    labour = rs11.getString("labour");
                    amt2 = rs11.getString("amt");

                }

                response.sendRedirect("quoteview.jsp");
            }
            catch(Exception e){
                e.printStackTrace();
            }
        }
        else{

            System.out.println("hhhhhhh");
        }
    }
}

```

```
}
```

```
}
```

```
}
```

```
import java.io.IOException;
```

```
import java.io.PrintWriter;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.PreparedStatement;
```

```
import java.util.Random;
```

```
import javax.servlet.RequestDispatcher;
```

```
import javax.servlet.ServletException;
```

```
import javax.servlet.annotation.WebServlet;
```

```
import javax.servlet.http.HttpServlet;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
```

```
import javax.servlet.http.HttpSession;
```

```
/**
```

```
 * Servlet implementation class Register
```

```

*/

@WebServlet("/Register")
public class Register extends HttpServlet {

    private static final long serialVersionUID = 1L;

    /**
     * Default constructor.
     */
    public Register() {
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        String n=request.getParameter("name");
        String p=request.getParameter("email");
        String e=request.getParameter("phone");
        String c=request.getParameter("password");

        Random r = new Random();
        Integer q =r.nextInt(1000);

        HttpSession session = request.getSession(true);
        session.setAttribute("uid", q);
    }
}

```

```

try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con=DriverManager.getConnection(
        "jdbc:mysql://localhost:3306/logreg","root","root");

    PreparedStatement ps=con.prepareStatement(
        "insert into reg values(?,?,?,?,?)");

    ps.setInt(1,q);
    ps.setString(2,n);
    ps.setString(3,p);
    ps.setString(4,e);
    ps.setString(5,c);

    int i=ps.executeUpdate();
    if(i>0){

        out.print("<html><body><script>alert('Register
Successfully');</script></body></html>");

        RequestDispatcher rs = request.getRequestDispatcher("login.html");
        rs.include(request, response);
    }

} catch (Exception e2) {System.out.println(e2);}

```

```
        out.close();
    }

}
```

```
import java.io.IOException;
import java.sql.PreparedStatement;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import utility.DB;

/**
 * Servlet implementation class updateu
 */
@WebServlet("/updateu")
public class updateu extends HttpServlet {
    private static final long serialVersionUID = 1L;
```

```

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    HttpSession ses = request.getSession();
    int id = (Integer)ses.getAttribute("uid");
    response.setContentType("text/html");
    String a13="PROCEED PAYMENT ON HOME PAGE";
    try {
        String sql=("update personreg set status='"+ a13 +" where uid=");
        int ps = DB.getconnection().prepareStatement(sql).executeUpdate();
        if(ps>0){
            response.sendRedirect("adminhome.html");
        }
    }

    catch(Exception e){
        e.printStackTrace();
    }

}}

```

```

import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;

```

```
import java.io.PrintWriter;
```

```
import java.sql.PreparedStatement;
```

```
import javax.imageio.ImageIO;
```

```
import javax.servlet.RequestDispatcher;
```

```
import javax.servlet.ServletException;
```

```
import javax.servlet.annotation.MultipartConfig;
```

```
import javax.servlet.annotation.WebServlet;
```

```
import javax.servlet.http.HttpServlet;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
```

```
import javax.servlet.http.HttpSession;
```

```
import javax.servlet.http.Part;
```

```
import utility.DB;
```

```
/**
```

```
 * Servlet implementation class uploadproof
```

```
 */
```

```
@WebServlet("/uploadproof")
```

```
@MultipartConfig(fileSizeThreshold = 1024 * 1024 * 2, maxFileSize = 1024 * 1024 * 10,  
maxRequestSize = 1024 * 1024 * 50)
```

```
public class uploadproof extends HttpServlet {
```

```
    private static final long serialVersionUID = 1L;
```

```
    /**
```

```

        * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
        */

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out= response.getWriter();

        final Part filePart = request.getPart("file");
        final Part filePart1 = request.getPart("file1");
        String d = request.getParameter("sno");

        InputStream fis = null;
        InputStream fis1 = null;// input stream of the upload file
        BufferedImage bImage = null;
        BufferedImage aImage = null;
        fis = filePart.getInputStream();
        fis1 = filePart1.getInputStream();
        bImage = ImageIO.read(fis);
        aImage = ImageIO.read(fis1);

        ImageIO.write(bImage, "jpg", new File("E:\\Solar\\WebContent\\proof\\"+ d
+"adh.jpg"));
        ImageIO.write(aImage, "jpg", new File("E:\\Solar\\WebContent\\proofs\\"+ d
+"land.jpg"));
        try {
            String sql=("update personreg set aadh=?,land=? where uid='"+ d +"'");

```

```

        PreparedStatement ps = DB.getconnection().prepareStatement(sql);

        ps.setString (1," proof/"+ d +"adh.jpg" );
        ps.setString (2," proofs/"+ d +"land.jpg" );
        int i = ps.executeUpdate();
        if(i>0)
        {

            response.sendRedirect("Orderpdf.jsp");
        }
        else
        {
            response.sendRedirect("adhar.jsp");
            System.out.println("insert unsuccessful");
        }
    }
    catch(Exception e1)
    {
        e1.printStackTrace();
    }
    out.close();
}

}

```

UTILITY:

```
package utility;
```

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DB {
    private static final String URL = "jdbc:mysql://localhost:3306/logreg";

    private static final String DRIVER = "com.mysql.jdbc.Driver";

    private static final String USERNAME = "root";

    private static final String PASSWORD = "root";

    private static Connection connection = null;

    public static Connection getConnection() throws SQLException {
        if (connection != null)
            return connection;
        else {
            try {
                Class.forName(DRIVER);
                connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
            } catch (ClassNotFoundException e) {
                e.printStackTrace();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```
return connection;
```

```
}
```

```
}
```

```
}
```

SYSTEM TESTING AND IMPLEMENTATION

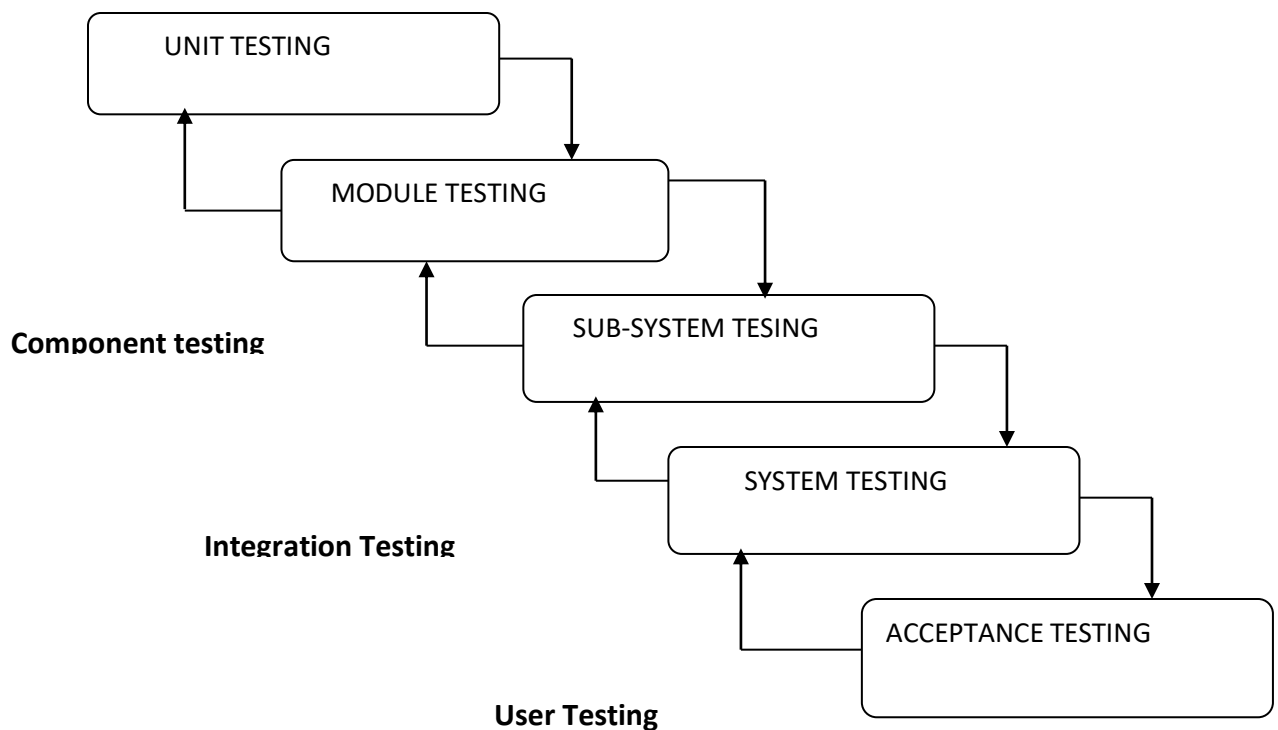
9.1. INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

9.2. STRATEGIC APPROACH TO SOFTWARE TESTING

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress is done by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally, we arrive at system testing, where the software and other system elements are tested as a whole.



9.3. Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

1. WHITE BOX TESTING

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds
- All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .We have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

2. BASIC PATH TESTING

The established technique of flow graph with Cyclamate complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graphs.

Determine the Cyclamate complexity of the resultant flow graph, using formula:

$$V(G) = E - N + 2 \text{ or}$$

$$V(G) = P + 1 \text{ or}$$

$$V(G) = \text{Number of Regions}$$

Where $V(G)$ is Cyclomatic complexity,

E is the number of edges,

N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

3. CONDITIONAL TESTING

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generated on particular condition is traced to uncover any possible errors.

4. DATA FLOW TESTING

This type of testing selects the path of the program, according to the location of the definition and use of variables. This kind of testing was used only when some local variable were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested statements.

5. LOOP TESTING

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

- All the loops were tested at their limits, just above them and just below them.
- All the loops were skipped at least once.
- For nested loop test the innermost loop first and then work outwards.
- For concatenated loops the values of dependent loops were set with the help of a connected loop.

No	Test Scenario	Expected Result	Test Result
1	Username is correct. Password is incorrect.	Username and Password is incorrect.	Username and Password is incorrect.
2	Username is incorrect. Password is correct.	Username and Password is incorrect.	Username and Password is incorrect.
3	Username is empty. Password is correct.	Username is required.	Username is required.
4	Username is correct. Password is empty.	Password is required.	Password is required

5	Both Username and Password is incorrect.	Username and Password is incorrect.	Username and Password is incorrect.
6	Both Username and Password is empty.	Username and Password is required.	Username and Password is required.
7	Both Username and Password is correct.	Login Successful.	Login Successful.

CHAPTER-10

SYSTEM SECURITY

10.1 INTRODUCTION

Security system can be divided into four related issues: The protection of computer based resources that includes hardware, software, data, procedures and people against unauthorized use or natural

Disaster is known as System Security.

- Security
- Integrity
- Privacy
- Confidentiality

SYSTEM SECURITY refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.

DATA SECURITY is the protection of data from loss, disclosure, modification and destruction.

SYSTEM INTEGRITY refers to the proper functioning of hardware and programs, appropriate physical security and safety against external threats such as eavesdropping and wiretapping.

PRIVACY defines the rights of the user or organizations to determine what information they are willing to share with or accept from others and how the organization can be protected against unwelcome, unfair or excessive dissemination of information about it.

CONFIDENTIALITY is a special status given to sensitive information in a database to minimize the possible invasion of privacy. It is an attribute of information that characterizes its need for protection.

10.2 SECURITY IN SOFTWARE

System security refers to various validations on data in the form of checks and controls to avoid the system from failing. It is always important to ensure that only valid data is entered and only valid operations are performed on the system. The system employs two types of checks and controls:

CLIENT SIDE VALIDATION

Various client side validations are used to ensure on the client side that only valid data is entered. Client side validation saves server time and load to handle invalid data. Some checks are imposed:

- JavaScript is used to ensure those required fields are filled with suitable data only. Maximum lengths of the fields of the forms are appropriately defined.
- Forms cannot be submitted without filling up the mandatory data so that manual mistakes of submitting empty fields that are mandatory can be sorted out at the client side to save the server time and load.
- Tab-indexes are set according to the need and taking into account the ease of use while working with the system.

SERVER SIDE VALIDATION

Some checks cannot be applied on the client side. Server side checks are necessary to save the system from failing and intimating the user that some invalid operation has been performed or the performed operation is restricted. Some of the server side checks imposed is:

- A server side constraint has been imposed to check for the validity of primary key and foreign key. A primary key value cannot be duplicated. Any attempt to duplicate the primary value results in a message intimating the user about those values through the forms using foreign key can be updated only of the existing foreign key values.
- The user is intimated through appropriate messages about the successful operations or exceptions occurring at server side.
- Various Access Control Mechanisms have been built so that one user may not agitate upon another. Access permissions to various types of users are controlled according to the

organizational structure. Only permitted users can log on to the system and can have access according to their category. User- name, passwords and permissions are controlled the server side.

- Using server side validation, constraints on several restricted operations are imposed.

CHAPTER-11

CONCLUSION

CONCLUSION:

we built up an AL system for assessing control-applicable sets, for example, reachable sets in a information driven way for frameworks we depend on testing and reenactments to coordinate the set assessment issue. By this project each and every variation of water level is informed to web server through internet & nearby people can be informed in time. Thus, saving lots of lives avoiding unpleasant scenarios . While using the filter method in this solar application, it will get the relevant information from the given input, so it could be help the people to know the type of product is suitable for their purpose. The filter method is used to give relevant information to the users (people) according what type of input is given. By the future work this application integrated with Artificial intelligence the user give the information they get better analyses data and to get more relevant live data information and also give the optimal suggestion to the user then the user will understand very easily. so create the world with the energy and save the natural resources for the future .