IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Defending Against Web Application Attacks: Approaches, Challenges and Implications

SHREENITHI.P, EVANGELINE CHRISTINA.D, HEMAPRIYA.Y, MRS.MANJULEESWARI
STUDENT, STUDENT, STUDENT, ASSISTANT PROFESSOR, B TECH INFORMATION TECHNOLOGY,
PANIMALAR ENGINEERING COLLEGE, CHENNAI, INDIA

Abstract : Probably the most hazardous web assaults, for example, Cross-Site Scripting and SQL infusion, misuse vulnerabilities in web applications that may acknowledge and process information of questionable starting point without appropriate approval or sifting, permitting the infusion and execution of dynamic or space explicit language code. These assaults have been always beating the arrangements of different security release suppliers in spite of the various countermeasures that have been proposed in the course of recent years. In this paper, we give an investigation on different guard systems against web code infusion assaults. We propose a model that features the key shortcomings empowering these assaults, and that gives a typical point of view to examining the accessible resistances. We at that point arrange and investigate a set of 41 recently proposed barriers dependent on their exactness, execution, arrangement, security, and accessibility qualities. Discovery exactness is of specific significance, as our discoveries show that numerous guard components have been tried in a poor way. Likewise, we see that a few components can be skirted by aggressors with information on how the instruments work. At long last, we talk about the consequences of our examination, with accentuation on factors that may thwart the across the board appropriation of resistances in practice.

SCOPE:

We provide an analysis on various defense mechanisms against web code injection attacks. We propose an exploitation model which highlights that most of the steps needed to mount different types of code injection attacks are common.

OBJECTIVE:

We see that attack prevention may take place either at the server or the client-side. A proxy is typically placed in front of the server to examine the server's responses before they reach a user's browser. A modified browser or a library that is securely downloaded from the server checks the responses for potential attacks.

EXISTING SYSTEM:

In Existing System, the web application attacks may involve security misconfigurations, broken authentication and session management, or other issues. Some of the most dangerous and prevalent web application attacks, however, exploit vulnerabilities associated with improper validation or filtering of untrusted inputs, resulting in the injection of malicious script or domain-specific language code. Attackers seem to find new ways to introduce malicious code to applications using a variety of languages and techniques. Meanwhile, during the last decade, there have been numerous mechanisms designed to detect one or more of types of such attacks. Although some deployed and widely used frameworks, such as CSP, share characteristics (for instance, HTML sanitization and eval handling) with previous proposals, most research works are still not used in practice.

DISADVANTAGE:

- Accuracy: protection mechanisms are as good as their detection capability; this requires low false positive and false negative rates.
- The increasing number of SQL injection attacks suggests that programmers are not always that careful.
- One of our key findings indicates that many proposed defenses are tested in a poor manner.

PROPOSED SYSTEM:

We explore how different attacks associated with the exploitation of untrusted input validation errors can be modeled under a common perspective. To that end, we propose an exploitation model which highlights that most of the steps needed to mount different types of code injection attacks are common. This is validated by the fact that some protection mechanisms defend against more than one of these types of attacks. We categorize a selection of representative protection mechanisms. In our selection we include protection mechanisms that counter web attacks when they take place. Similarly, dynamic analysis techniques that examine applications to identify vulnerabilities that may lead to the attacks

ADVANTAGE:

- We provide a unified exploitation model for different types of web application attacks based on code injection.
- We categorize and analyze proposed defenses using a set of criteria that are important for building protection mechanisms.
- We put emphasis on factors that may hinder the widespread deployment of protection mechanisms, and the transition of tools from research to practice.

SYSTEM CONFIGURATION:

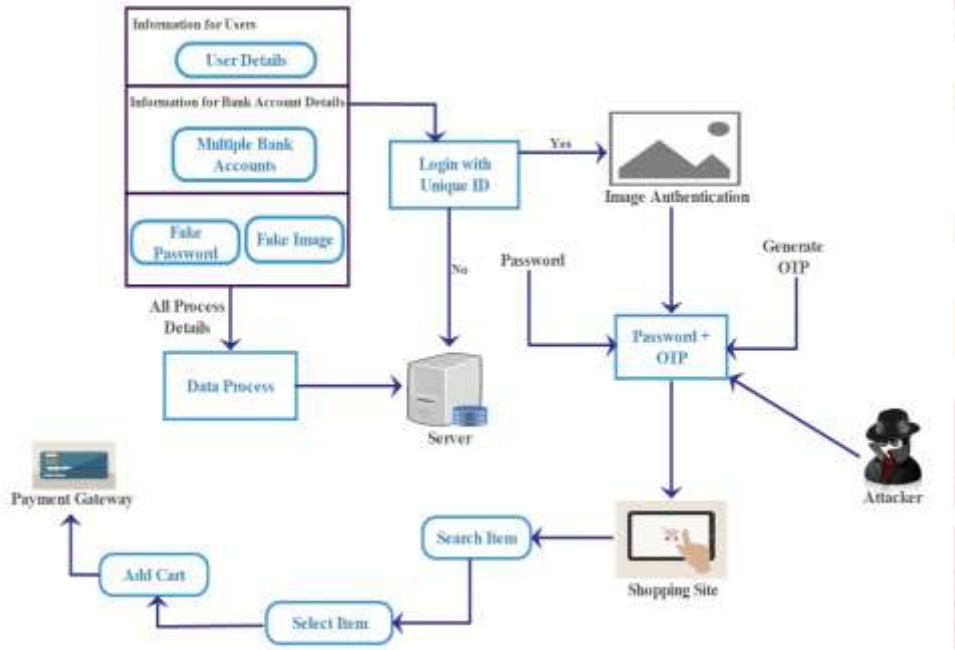
H/W SYSTEM CONFIGURATION:

- Processor - Intel Pentium
- RAM - 4 GB
- Hard Disk - 260 GB

S/W SYSTEM CONFIGURATION:

- Operating System - Windows 7/8/10
- Front End - JAVA
- Back End - MySQL
- Tools - NetBeans 7.3.1

Architecture/Block Diagram:



MODULES

- Adversary Models.
- Session-Key-Extraction.
- Security Flaws with the Session-Key.
- Current-Password-Extraction.
- Password-Changing Phase.
- Countermeasures.

1. Adversary Models.

Intuitively, an attacker on a smart-card-based password authentication protocol should be unable to make successful log-in only with the smart card (or the password),or compromise other additional properties (key agreement).To capture these requirements,we define the potential attacker from two aspects,namely the behavior of the attacker and the information compromised by the attacker.As an interactive protocol,a smart-card-based password authentication protocol may be faced with a passive attacker and an active attacker A passive attacker can obtain messages transmitted between users and the server.This is due to the fact that communication channels are generally insecure,and the attacker can observe messages by eavesdropping.A passive attacker cannot interact with any of the parties in smart-card-based password authentication protocols In addition to message eavesdropping,an active attacker can also inject andmodifymessages in the communication between the user and the server.In particular,the attacker can initiate a log-in request on behalf of the user,or act as the server by sending messages to the user.An active attacker can also request any session keys adaptively (if the protocol supports key agreement).It is evident that an active attacker is more powerful than a passive attacker.

2. Session-Key-Extraction.

We first show that a passive attacker with smart card can calculate the session key between the server and the user in the protocol proposed.At the end of the log-in phase the session key between the user and the server is It suffices to compute Sk and u.stored in the smart card before the log-in phase.More precisely,Vi is generated in the registration phase,and c is added to the memory of the smart card in the pre-computation phase.The purpose of pre-computation is to speed up the computation in the authentication,which should be regarded as a separate phase from the log-in phase.Thus,to reduce the computational load in log-in phase in,the smart card must complete the calculation of before the log-in phase,rather than performing the calculation at the beginning of the log-in phase.where the computational cost in the log-in phase does not include the calculation of the attacker can obtain if he/she can extract the information in the smart card before the log-in phase.It remains to show that the adversary can obtain u as well.the sever sends to the smart card, which enables an eavesdropping (passive) adversary to obtain u.This completes the description of how a passive attacker with smart card can obtain and calculate the session key.

3. Security Flaws with the Session-Key

We now show that a successful attack against the session key will undermine the security of whole system from at least two aspects.First,the communication between the user and the server is no longer secure the purpose of the session is to establish a secure communication between the user and the server.The communication,thus,will not be secure if Sk is compromised.As an example,an adversary with Sk is able to decrypt any ciphertexts which are generated using the encryption key Sk.Message authentication could also fail if it solely relies on Sk.Second,the adversary can freely change the user's password.The given attack is different from the common offlinedictionary attack with the smart card,as the adversary does not guess the user's password.In other words,the adversary does not have the user's password at the end of the attack.This,however,cannot prevent a successful login from the attacker on behalf of the user,since the adversary is able to change the password with the session key Sk.Once the log-in phase is completed,the adversary can immediately invoke the password-changing phase,namely the adversary chooses a new pair.Let the response from the server be which can be decrypted by the adversary with the session key Sk.After that,the adversary can successfully login to the server (on behalf of the user with identity) with the new password i and the new smart card .

4. Current-Password-Extraction.

With smart card,we show a passive attacker can also successfully find the user's current password using offlinedictionary attack.the smart card sends the server a message.This message can assist an offline-dictionary attacker to verify if the guess of the password is correct.The calculation of MU involves,among which only PWi is unknown to an eavesdropping adversary with the smart card. are stored in the smart card before the log-in phase, and u is sent to the smart card by the server in the log-in phase.for how a passive attacker with smart card can obtain.The adversary can try each possible word pw in the password dictionary until he/she finds a password matches Such a password will be PWi if the hash function h is collision-resistent, and thus the offline-dictionary attack with smart card is successful. This completes our analysis of Juang-Chen-Liaw's scheme.

5. Password-Changing Phase.

We note that such an adversary is stronger than that considered ,where the adversary can obtain the information in the smart card but only once. If the adversary can capture the information in the smart card once, we believe the adversary can also do it for the second time. As an example, one can obtain the information in the smart card via an illegal card reader. This could occur more than once without the awareness of the smart card owner (e.g., the attacker could steal the smart card and send it back after extracting the data stored in the smart card). In the above attacking scenario, the other assumption is that the user will change the password at least twice. We believe this is also a reasonable assumption as changing password on a regular basis has been regarded as one of good password habits. This completes the description of the attacking scenario we are concerned about, which we believe falls into the category of passive attacker with smart card defined. It remains to show how to extract the two passwords.

6. Countermeasures.

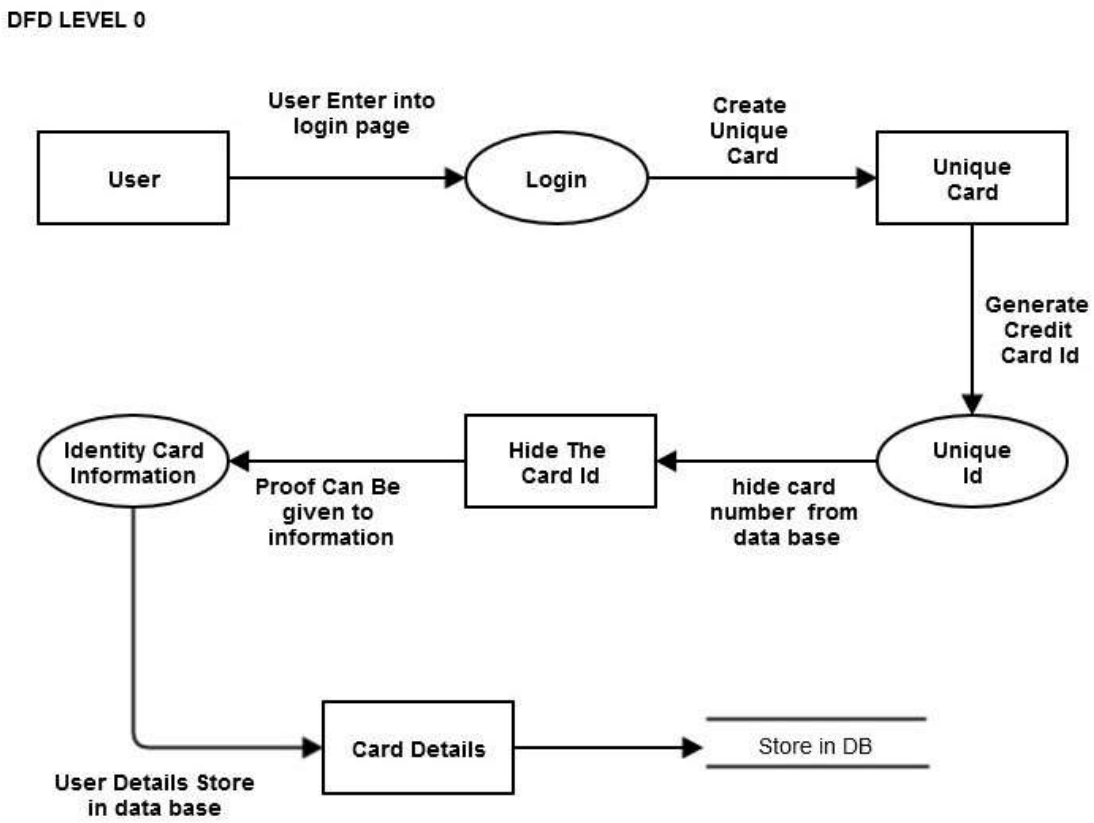
This section shall brief the countermeasures to thwart the security threats discovered in previous sections.Juang-Chen-Liaw's scheme:The analysis has shown that the scheme is secure against offline-dictionary attack with the smart card.This,however,is under the assumption that the adversary only has the information generated in the registration phase,rather than all data stored in the smart card.In this sense,our attacks do not show anything wrong in the analysis,as adversaries considered in this section are stronger and more powerful than those.(Again, the described attacks in this section reflect the potential threats in real life,where adversaries are able to obtain all data stored in the smart card).In the pre-computation phase,the smart card calculates c and e,where c is required

in both attacks.Without c,it is virtually infeasible to calculate the session key or verify the guess of the password using.Thus,neither of the attacks would be successful.To make the protocol secure against the given attacks, one can assume that the pre-computation phase is carried out in a secure manner and the output is kept secret from the adversary.Under this assumption,an adversary with the smart card is not able to obtain the information generated in the pre-computation phase (which is also stored in the smart card).Nevertheless,an adversary with the smart card can have the information generated in the registration phase.We observe that a trivial solution,only encrypting u with and sending cannot fix the problem.In fact,this will enable an offline-dictionary adversary with the smart card to verify the guess of the password.

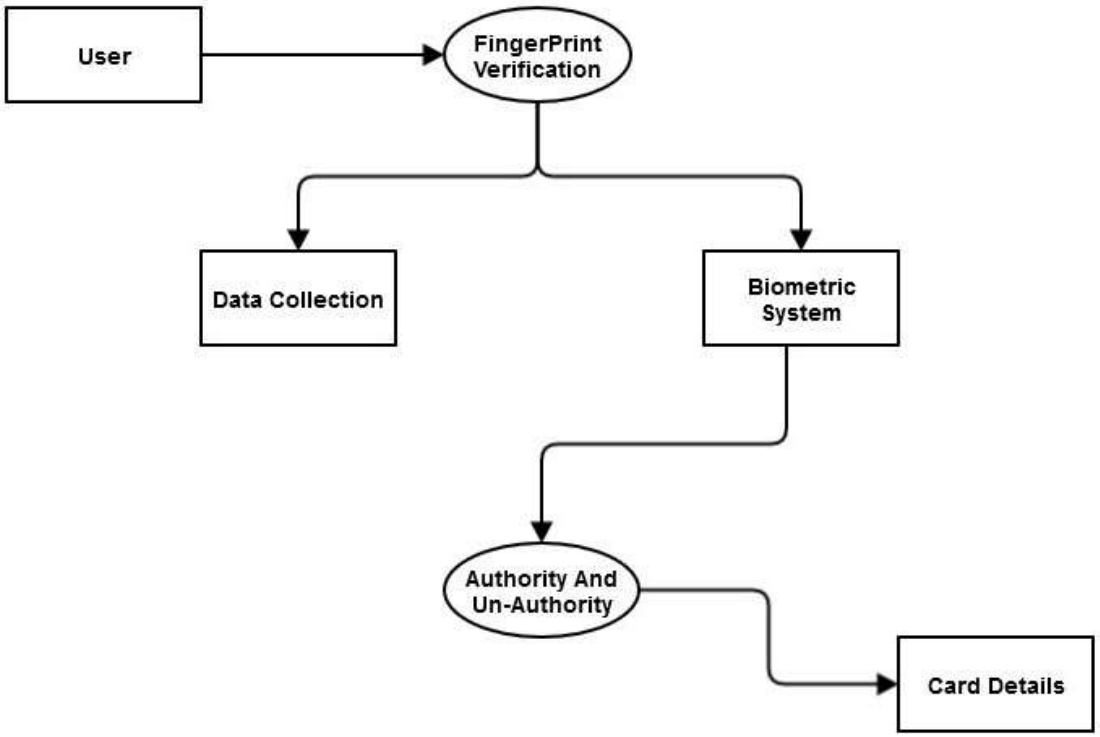
DATA FLOW DIAGRAM:

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of data processing. The DFD is designed to show how a system is divided into smaller portions and to highlight the flow of data between those parts.

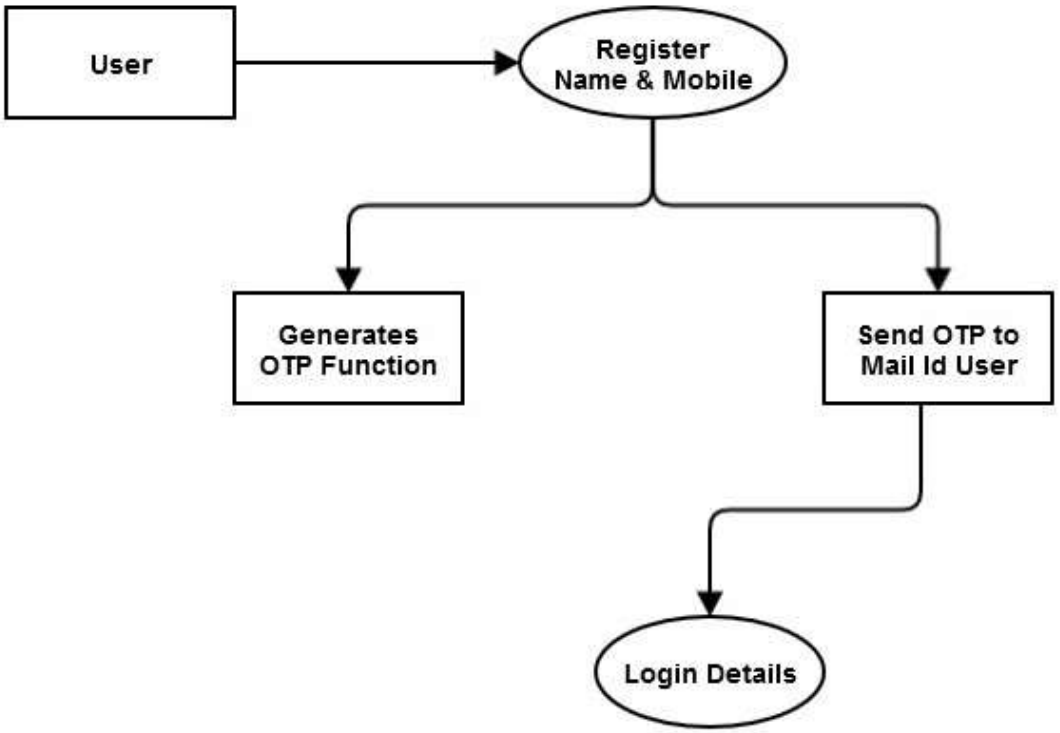
Data Flow Diagram (DFD) is an important technique for modeling a system's high-level detail by showing how input data is transformed to output results through a sequence of functional transformations.DFDs reveal relationships among and between the various components in a program or system. DFD consists of four major components: entities, processes, data stores and data flow.



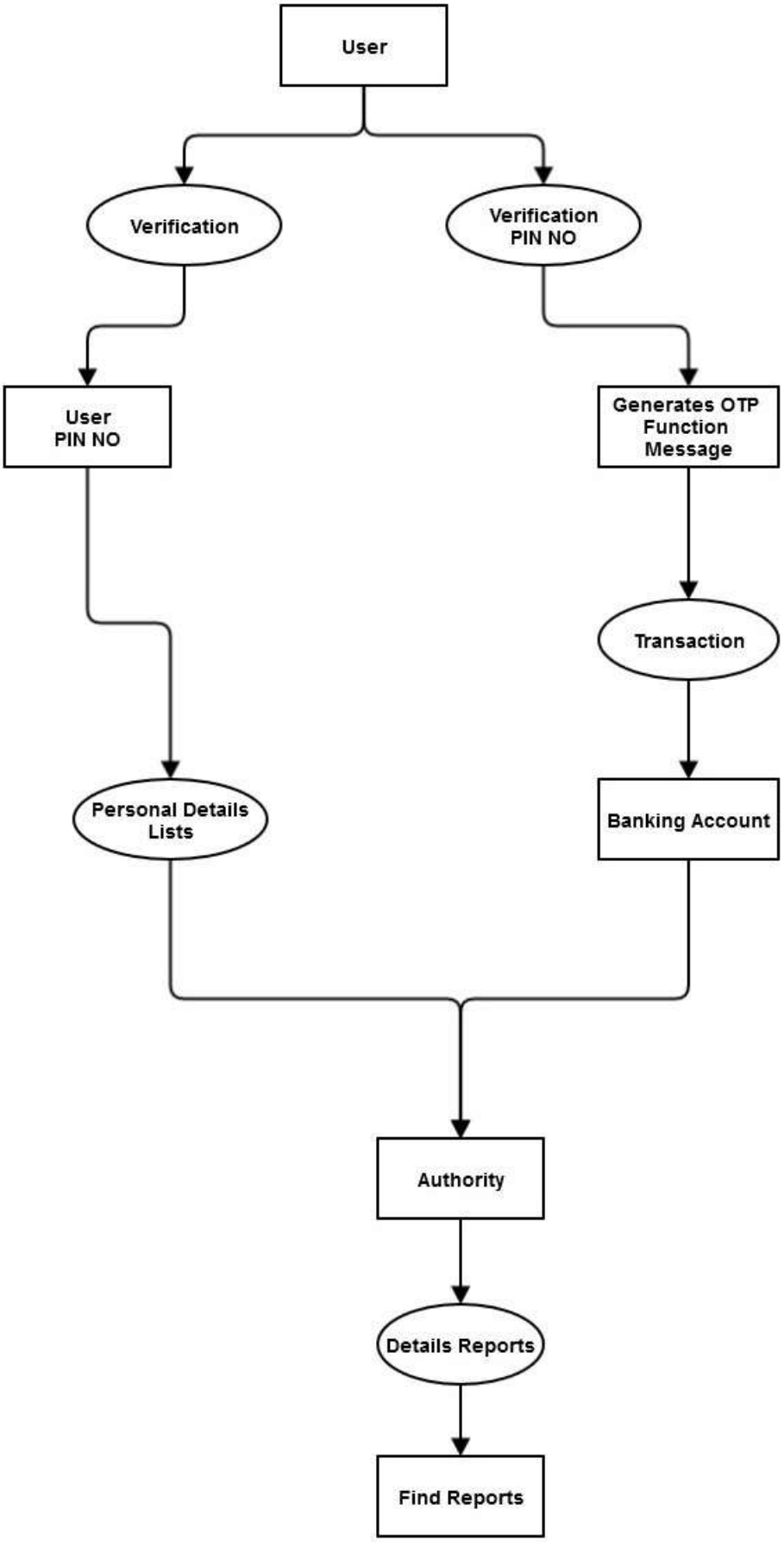
DFD LEVEL 1



DFD LEVEL 2

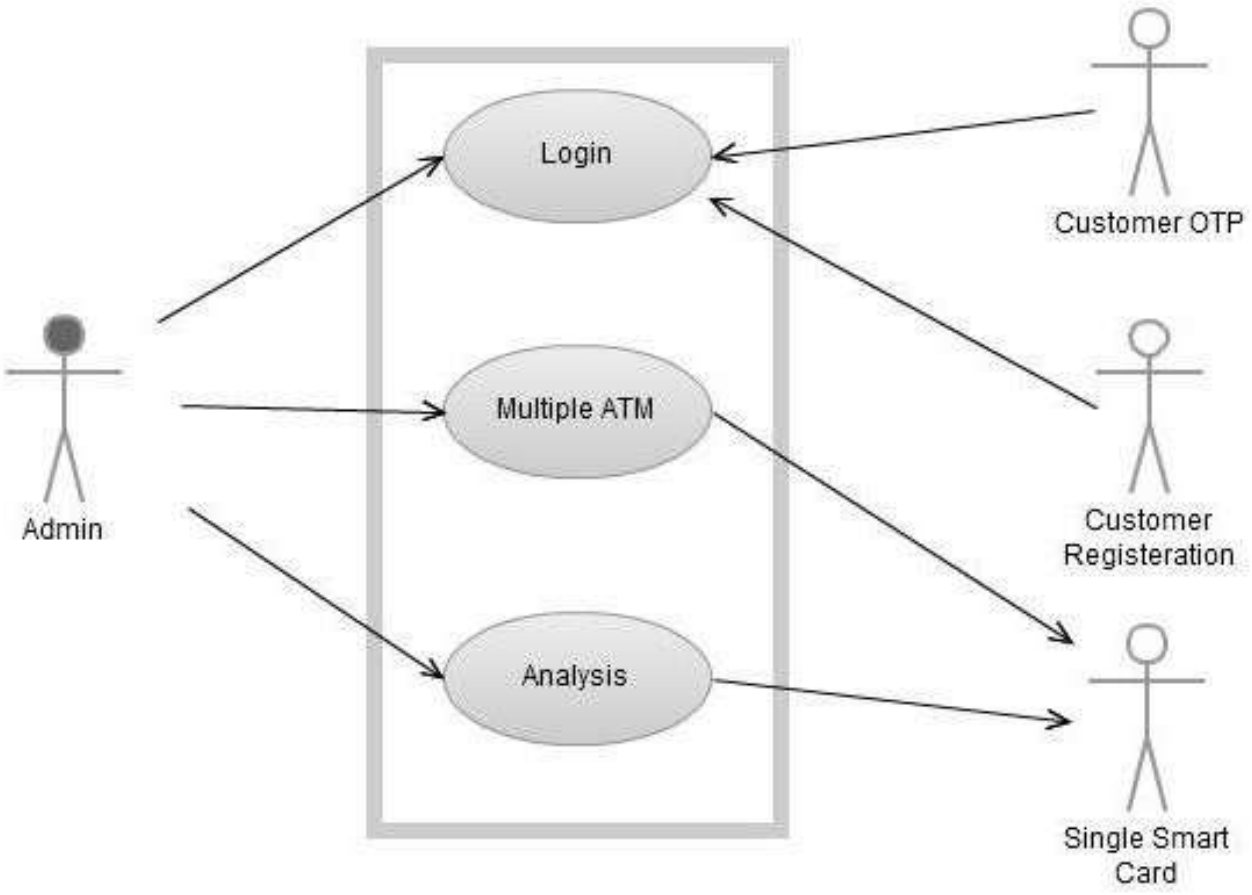


Over All DFD LEVEL



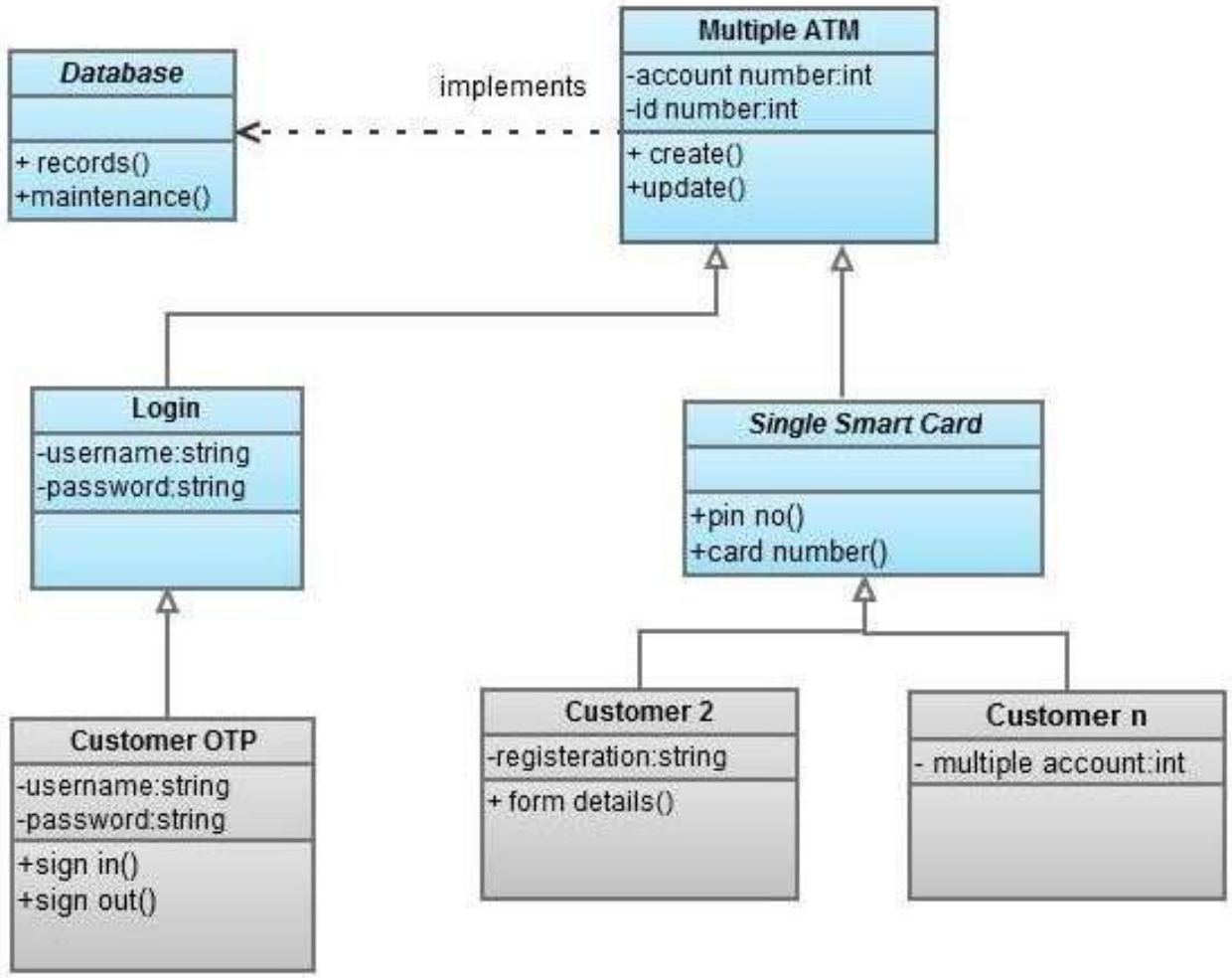
USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. Use case diagrams are formally included in two modeling languages defined by the OMG: the Unified Modeling Language (UML) and the Systems Modeling Language (SysML). Use case diagrams are behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.



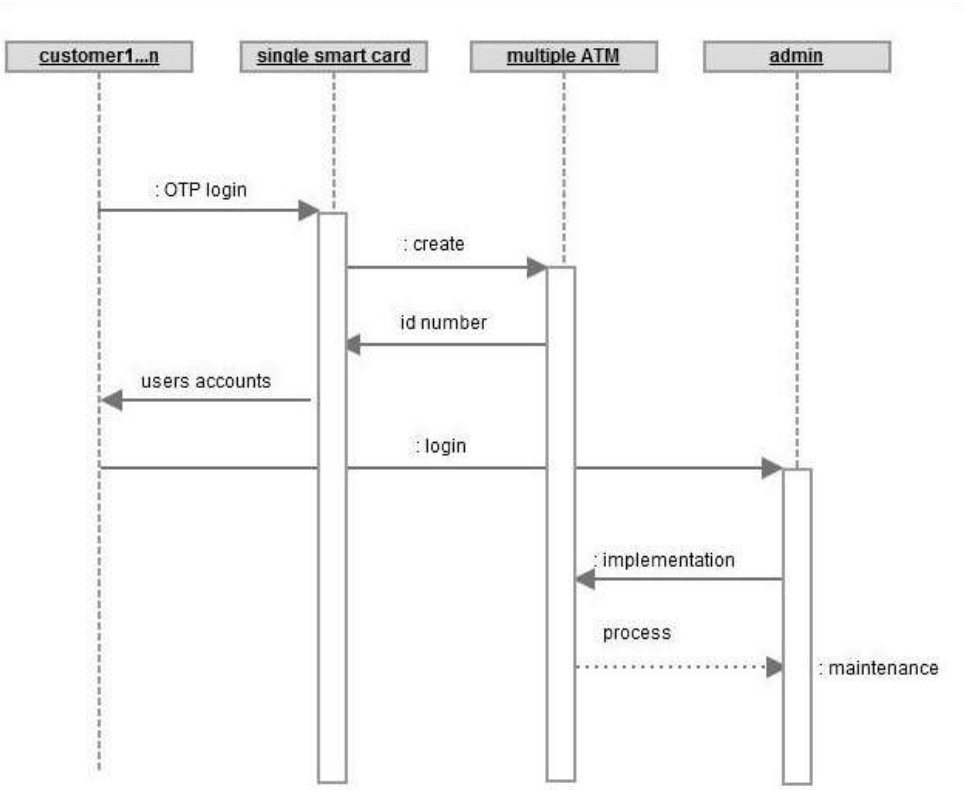
CLASS DIAGRAM:

A class diagram in the UML is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. Private visibility hides information from anything outside the class partition. Public visibility allows all other classes to view the marked information. Protected visibility allows child classes to access information they inherited from a parent class. In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In a class diagram, the classes are arranged in groups that share common characteristics. A class diagram resembles a flowchart in which classes are portrayed as boxes, each box having three rectangles inside. The top rectangle contains the name of the class; the middle rectangle contains the attributes of the class; the lower rectangle contains the methods, also called operations, of the class. Lines, which may have arrows at one or both ends, connect the boxes. These lines define the relationships, also called associations, between the classes.



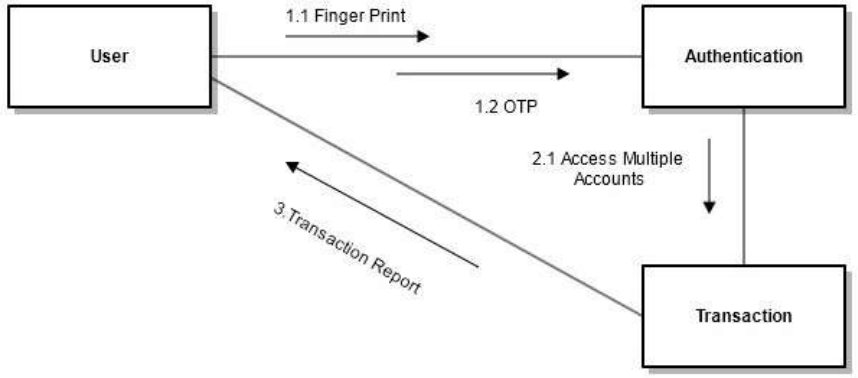
SEQUENCE DIAGRAM:

A sequence diagram in UML is a kind of interaction diagram that shows how processes operate with one another and in what order. A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams typically are associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



COLLABORATION DIAGRAM:

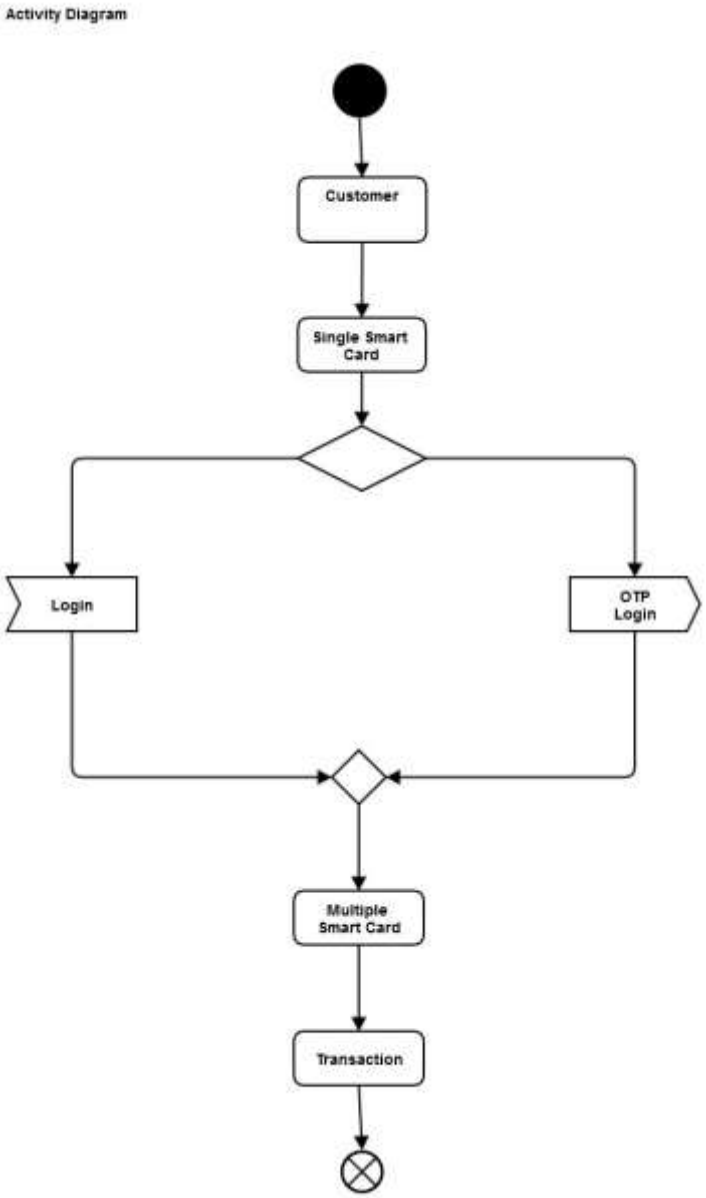
A collaboration diagram show the objects and relationships involved in an interaction, and the sequence of messages exchanged among the objects during the interaction. The collaboration diagram can be a decomposition of a class, class diagram, or part of a class diagram. it can be the decomposition of a use case, use case diagram, or part of a use case diagram. The collaboration diagram shows messages being sent between classes and object (instances). A diagram is created for each system operation that relates to the current development cycle (iteration).*A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among* software objects *in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.* A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time. Objects are shown as rectangles with naming labels inside. These labels are preceded by colons and may be underlined. The relationships between the objects are shown as lines connecting the rectangles. The messages between objects are shown as arrows connecting the relevant rectangles along with labels that define the message sequencing.



ACTIVITY DIAGRAM:

Activity diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. UML, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. UML activity diagrams could potentially model the internal logic of a complex operation. In many ways UML activity diagrams are the object-oriented equivalent of flow charts and data flow diagrams (DFDs) from structural development. Activity diagrams are graphical representations of workflows of stepwise activities and actions with support

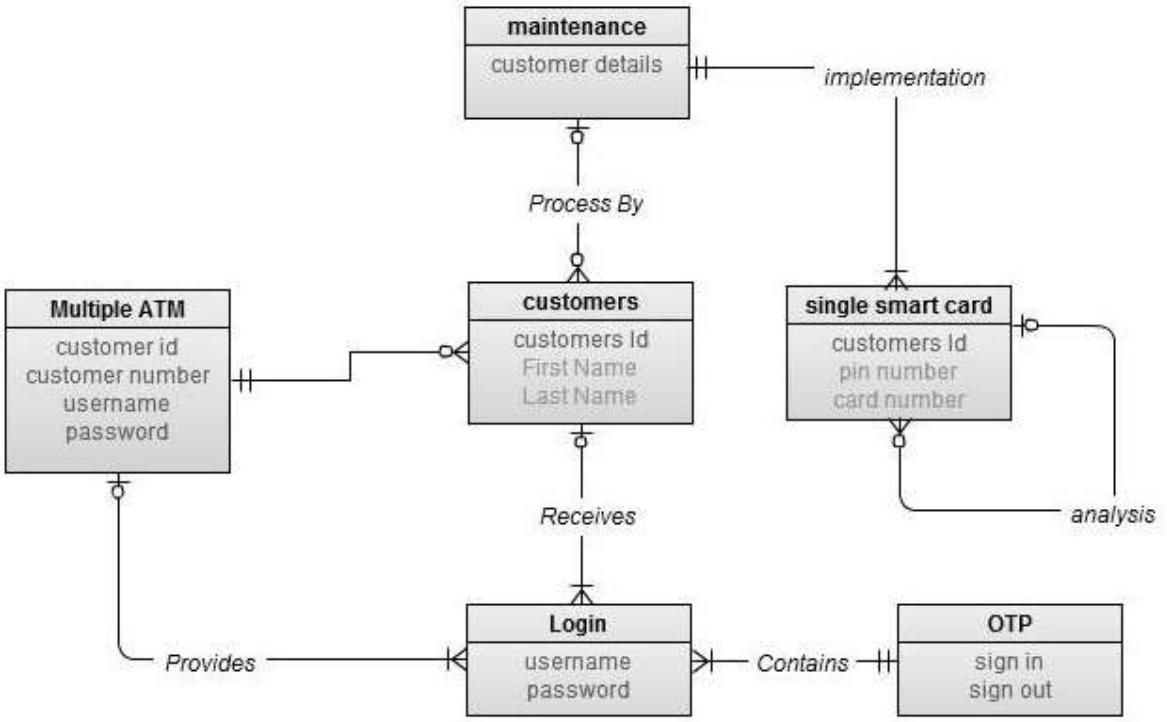
for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



E-R DIAGRAM:

In software engineering, an entity-relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called entity-relationship diagrams, ER diagrams, or ERDs.

An entity-relationship (ER) diagram is a specialized graphic that illustrates the relationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.



LITERATURE SURVEY:

Paper1

Title: Proxy-Terms Based Query Obfuscation Technique for Private Web Search

Author: SHARIQ BASHIR 1, DAPHNE TECK CHING LAI 2, AND OWAIS AHMED MALIK 2

Abstract:

Search engines store users' queries in a query log for performing personalized information retrieval. However, query logs cause privacy concerns and reveal a lot of information about individuals if used against them. Private web search (PWS) provides a privacy-preserving information retrieval (IR) facility which allows users to retrieve information from an IR system without revealing true search queries. Current PWS techniques that are explored in the domain of web search are query obfuscation-based private web search (OB-PWS). These techniques achieve web privacy by injecting cover queries into the user profiles. However, existing OB-PWS techniques submit true queries along with cover queries and achieve query obfuscation in an isolated manner without considering the similarity between consecutive queries. In this article, we propose a proxy-terms based query obfuscation technique that allows users to retrieve information from an IR system through proxy queries without submitting true queries. IR system automatically generates cover queries and true queries from the proxy queries and cannot differentiate whether the user is trying to retrieve information for the cover queries or true query. We analyzed the effectiveness of the proposed technique on test queries, and develop a similarity metric for testing the accuracy of the proposed technique. Promising results of experiments confirm the effectiveness of the proposed technique.

Year: 2022

Paper2

Title: Multi-Group ObScure Logging (MG-OSLo) A Privacy-Preserving Protocol for Private Web Search

Author: MOHIB ULLAH 1,2, RAFIULLAH KHAN 1,2, MUHAMMAD INAM UL HAQ3, ATIF KHAN 4, WAEL ALOSAIMI 5, MUHAMMAD IRFAN UDDIN 6, AND ABDULLAH ALHARBI5

Abstract:

The Web Search Engine (WSE) is a software system used to retrieve data from the web successfully. WSE uses the user's search queries to build the user's profile and provide personalized results. Users' search queries hold identifiable information that could compromise the privacy of the respective user. This work proposes a multi-group distributed privacy-preserving protocol (MG-OSLo) and tries to investigate the state-of-the-art distributed privacy-preserving protocols for computing web search privacy. The MG-OSLo comprises multiple groups in which each group has a fixed number of users. The MG-OSLo measures the impact of the multi-group on the user's privacy. The primary objective of this work is to assess local privacy and profile privacy. It aims at evaluating the impact of group size and group count on a user's privacy. Two grouping approaches are used to group the users in MG-OSLo, i.e. a non-overlapping group design and overlapping group design. The local privacy results reveal that the probability of linking a query to the user depends on the group size and group count. The higher the group size or group count, the lower the likelihood of relating the query to the user. The profile privacy computes the profile obfuscation level using a privacy metric Profile Exposure Level (PEL). Different experiments have been performed to compute the profile privacy of the subset of an AOL query

log for two situations: i) self-query submissions allowed and ii) self-query submissions not allowed. The privacy achieved by MG-OSLo is compared with the modern privacy-preserving protocol UUP(e), OSLo, and Co-utile protocols. The results show that the MG-OSLo provided better results as compared to OSLo, UUP, and Co-utile. Similarly, the multi-group has a positive impact on local privacy and user profile privacy.

Year: 2021

Paper3

Title: The Anonymity of the Dark Web: A Survey

Author: JAVERIAH SALEEM , RAFIQUL ISLAM , (Senior Member, IEEE),
AND MUHAMMAD ASHAD KABIR

Abstract:

The dark web is a section of the Internet that is not accessible to search engines and requires an anonymizing browser called Tor. Its hidden network and anonymity pave the way for illegal activities and help cybercriminals to execute well-planned, coordinated, and malicious cyberattacks. Cyber security experts agree that online criminal activities are increasing exponentially, and they are also becoming more rampant and intensified. These illegal cyber activities include various destructive crimes that may target a single person or a whole nation, for example, data breaches, ransomware attacks, black markets, mafias, and terrorist attacks. So, maintaining data privacy and secrecy is the new dilemma of the era. This paper has extensively reviewed various attacks and attack patterns commonly applied in the dark web. We have also classified these attacks in our unique trilogies classification system. Furthermore, a detailed overview of existing threat detection techniques and their limitations is discussed for anonymity providing services like Tor, I2P, and Freenet. Finally, the paper has identified significant weaknesses that make the dark web vulnerable to different attacks.

Year: 2022

Paper4

Title: Tail Time Defense Against Website Fingerprinting Attacks

Author: JINGYUAN LIANG 1, CHANSU YU 1, (Senior Member, IEEE), KYOUNGWON SUH2, (Member, IEEE), AND HYOIL HAN

Abstract:

In the past few years, many defense mechanisms have been proposed against website fingerprinting attacks. Walkie-Talkie (WT) built on top of the Tor network is known to be one of the most effective defense mechanisms. However, we observed that WT significantly increases the page loading time (time overhead) although the bandwidth overhead is not high. This paper analyzes the cause of the increased page loading time and presents a defending approach called Tail Time (TT), which addresses the problem by limiting the maximum time for which a pending request can block subsequent requests. Our experimental results indicate that the proposed TT defense can significantly reduce the page loading time while keeping the defense performance on par.

Year: 2022

Paper5

Title: Optimal Filter Assignment Policy Against Distributed Denial-of-Service Attack

Author: Rajorshi Biswas and Jie Wu Department of Computer and Information Sciences Temple University, Philadelphia,

Abstract:

A distributed denial-of-service (DDoS) attack is a cyber-attack in which attackers from different locations send out many requests to exhaust the capacity of a server. Current DDoS attack protection services filter out the DDoS attack packets in the middle of the path from the attacker to the servers. Some of the DDoS protection systems filter them out at the victim server. As a result, unnecessary attack traffic congests the network and wastes bandwidth. This can be minimized if we block them as early as possible. In this paper, we propose a DDoS attack protection system by using the filter router. The victim needs to wisely select and send filters to a subset of filter routers to minimize attack traffic and blockage of legitimate users (LUs). Many filters can easily minimize the attack traffic and blockage of LUs, but it is costly to the victim. So, we formulate two problems with different settings for selecting filter routers given a constraint on the number of filters. We propose dynamic programming solutions for both problems. Both problems consider the blockage of all attack traffic before it reaches the victim. We conduct extensive simulation to support our solutions.**Year: 2020**

SOFTWARE ENVIRONMENT:

1. **Java Technology**

Java technology is both a programming language and a platform.

1. The Java Programming Language
2. The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple

- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java byte codes* —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.

You can think of Java byte codes as the machine code instructions for the *Java Virtual Machine* (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make "write once, run anywhere" possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

3. **The Java Platform**

A *platform* is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The *Java Virtual Machine* (Java VM)
- The *Java Application Programming Interface* (Java API)

You've already been introduced to the Java VM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as *packages*. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.

Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

2. ***What Can Java Technology Do?***

The most common types of programs written in the Java programming language are *applets* and *applications*. If you've surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs.

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a *server* serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a *servlet*. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

- **The essentials**: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
- **Applets**: The set of conventions used by applets.
- **Networking**: URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses.
- **Internationalization**: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.
- **Security**: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.
- **Software components**: Known as JavaBeans™, can plug into existing component architectures.
- **Object serialization**: Allows lightweight persistence and communication via Remote Method Invocation (RMI).
- **Java Database Connectivity (JDBC™)**: Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in the Java 2 SDK.

3. How Will Java Technology Change My Life?

We can't promise you fame, fortune, or even a job if you learn the Java programming language. Still, it is likely to make your programs better and requires less effort than other languages. We believe that Java technology will help you do the following:

- **Get started quickly**: Although the Java programming language is a powerful object-oriented language, it's easy to learn, especially for programmers already familiar with C or C++.
- **Write less code**: Comparisons of program metrics (class counts, method counts, and so on) suggest that a program written in the Java programming language can be four times smaller than the same program in C++.
- **Write better code**: The Java programming language encourages good coding practices, and its garbage collection helps you avoid memory leaks. Its object orientation, its JavaBeans component architecture, and its wide-ranging, easily extendible API let you reuse other people's tested code and introduce fewer bugs.
- **Develop programs more quickly**: Your development time may be as much as twice as fast versus writing the same program in C++. Why? You write fewer lines of code and it is a simpler programming language than C++.
- **Avoid platform dependencies with 100% Pure Java**: You can keep your program portable by avoiding the use of libraries written in other languages. The 100% Pure Java™ Product Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.
- **Write once, run anywhere**: Because 100% Pure Java programs are compiled into machine-independent byte codes, they run consistently on any Java platform.
- **Distribute software more easily**: You can upgrade applets easily from a central server. Applets take advantage of the feature of allowing new classes to be loaded "on the fly," without recompiling the entire program.

1. ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a *de facto* standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas

the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when you setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE. There is a 16-bit and a 32-bit version of this program and each maintains a separate list of ODBC data sources.

From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn't change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn't as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of "plug-in" database connectivity modules, or *drivers*. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

To gain a wider acceptance of JDBC, Sun based JDBC's framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after.

The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

JDBC Goals

Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java.

The goals that were set for JDBC are important. They will give you some insight as to why certain classes and functionalities behave the way they do. The eight design goals for JDBC are as follows:

SQL Level API

The designers felt that their main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently. Attaining this goal allows for future tool vendors to "generate" JDBC code and to hide many of JDBC's complexities from the end user.

SQL Conformance

SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors,

JDBC will allow any query statement to be passed through it to the underlying database driver. This allows the connectivity module to handle non-standard functionality in a manner that is suitable for its users.

JDBC must be implemental on top of common database interfaces

The JDBC SQL API must "sit" on top of other common SQL level APIs. This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and vice versa.

Provide a Java interface that is consistent with the rest of the Java system

Because of Java's acceptance in the user community thus far, the designers feel that they should not stray from the current design of the core Java system.

Keep it simple

This goal probably appears in all software design goal listings. JDBC is no exception. Sun felt that the design of JDBC should be very simple, allowing for only one method of completing a task per mechanism. Allowing duplicate functionality only serves to confuse the users of the API.

Use strong, static typing wherever possible

Strong typing allows for more error checking to be done at compile time; also, less error appear at runtime.

Keep the common cases simple

Because more often than not, the usual SQL calls used by the programmer are simple SELECT's, INSERT's, DELETE's and UPDATE's, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

Finally we decided to proceed the implementation using Java Networking.

And for dynamically updating the cache table we go for MS Access database.

Java ha two things: a programming language and a platform.

Java is a high-level programming language that is all of the following

Simple Architecture-neutral

Object-oriented Portable

Distributed High-performance

nterpreted multithreaded

Robust Dynamic

Secure

Java is also unusual in that each Java program is both compiled and interpreted. With a compile you translate a Java program into an intermediate language called Java byte codes the platform-independent code instruction is passed and run on the computer.

Compilation happens just once; interpretation occurs each time the program is executed. The figure illustrates how this works.

You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of the Java VM. The Java VM can also be implemented in hardware.

Java byte codes help make "write once, run anywhere" possible. You can compile your Java program into byte codes on my platform that has a Java compiler. The byte codes can then be run any implementation of the Java VM. For example, the same Java program can run Windows NT, Solaris, and Macintosh.

Networking:-

TCP/IP stack

1.TCP/IP stack is shorter than the OSI one:

TCP is a connection-oriented protocol; UDP (User Datagram Protocol) is a connectionless protocol.

IP datagram's

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

UDP

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client/server model - see later.

TCP

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

Internet addresses

In order to use a service, you must be able to find it. The Internet uses an address scheme for machines so that they can be located. The address is a 32 bit integer which gives the IP address. This encodes a network ID and more addressing. The network ID falls into various classes according to the size of the network address.

Network address

Class A uses 8 bits for the network address with 24 bits left over for other addressing. Class B uses 16 bit network addressing. Class C uses 24 bit network addressing and class D uses all 32.

Subnet address

Internally, the UNIX network is divided into sub networks. Building 11 is currently on one sub network and uses 10-bit addressing, allowing 1024 different hosts.

Host address

8 bits are finally used for host addresses within our subnet. This places a limit of 256 machines that can be on the subnet.

otal address

The 32 bit address is usually written as 4 integers separated by dots.

Port addresses

A service exists on a host, and is identified by its port. This is a 16 bit number. To send a message to a server, you send it to the port for that service of the host that it is running on. This is not location transparency! Certain of these ports are "well known".

Sockets:-

A socket is a data structure maintained by the system to handle network connections. A socket is created using the call socket. It returns an integer that is like a file descriptor. In fact, under Windows, this handle can be used with Read File and Write File functions.

```
#include <sys/types.h>

#include <sys/socket.h>

int socket(int family, int type, int protocol);
```

Here "family" will be AF_INET for IP communications, protocol will be zero, and type will depend on whether TCP or UDP is used. Two processes wishing to communicate over a network create a socket each. These are similar to two ends of a pipe - but the actual pipe does not yet exist.

JFree Chart

JFreeChart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. JFreeChart's extensive feature set includes:A consistent and well-documented API, supporting a wide range of chart types; A flexible design that is easy to extend, and targets both server-side and client-side applications; Support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG); JFreeChart is "open source" or, more specifically, free software. It is distributed under the terms of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications.

Map Visualizations

Charts showing values that relate to geographical areas. Some examples include: (a) population density in each state of the United States, (b) income per capita for each country in Europe, (c) life expectancy in each country of the world. The tasks in this project include:Sourcing freely redistributable vector outlines for the countries of the world, states/provinces in particular countries (USA in particular, but also other areas); Creating an appropriate dataset interface (plus default implementation), a rendered, and integrating this with the existing XYPlot class in JFreeChart; Testing, documenting, testing some more, documenting some more.

Time Series Chart Interactivity

Implement a new (to JFreeChart) feature for interactive time series charts --- to display a separate control that shows a small version of ALL the time seriesdata, with a sliding "view" rectangle that allows you to select the subset of the time series data to display in the main chart.

Dashboards

There is currently a lot of interest in dashboard displays. Create a flexible dashboard mechanism that supports a subset of JFreeChart chart types (dials, pies, thermometers, bars, and lines/time series) that can be delivered easily via both Java Web Start and an applet.

Property Editors

The property editor mechanism in JFreeChart only handles a small subsetof the properties that can be set for charts. Extend (or reimplement) this mechanism to provide greater end-user control over the appearance of the charts.

What is Database?

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching, and replicating the data it holds.

Other kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those type of systems.

So now a days we use relational database management systems (RDBMS) to store and manager huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys.

A **Relational DataBase Management System (RDBMS)** is a software that:

- Enables you to implement a database with tables, columns, and indexes.
- Guarantees the Referential Integrity between rows of various tables.
- Updates the indexes automatically.

- Interprets an SQL query and combines information from various tables.

RDBMS Terminology:

Before we proceed to explain MySQL database system, lets revise few definitions related to database.

- **Database:** A database is a collection of tables, with related data.
- **Table:** A table is a matrix with data. A table in a database looks like a simple spreadsheet.
- **Column:** One column (data element) contains data of one and the same kind, for example the column postcode.
- **Row:** A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
- **Redundancy:** Storing data twice, redundantly to make the system faster.
- **Primary Key:** A primary key is unique. A key value can not occur twice in one table. With a key you can find at most one row.
- **Foreign Key:** A foreign key is the linking pin between two tables.
- **Compound Key:** A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
- **Index:** An index in a database resembles an index at the back of a book.
- **Referential Integrity:** Referential Integrity makes sure that a foreign key value always points to an existing row.

MySQL Database:

MySQL is a fast, easy-to-use RDBMS used being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons.

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open source GPL license allows programmers to modify the MySQL software to fit their own specific environments

SOFTWARE REQUIREMENT SPECIFICATION

REQUIREMENT SPECIFICATION:

INTRODUCTION:

Purpose: The main purpose for preparing this document is to give a general insight into the analysis and requirements of the existing system or situation and for determining the operating characteristics of the system.
Scope: This Document plays a vital role in the development life cycle (SDLC) as it describes the complete requirement of the system. It is meant for use by the developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.
Developers Responsibilities Overview:
The developer is responsible for:

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

Functional Requirements:
Inputs:
The major inputs for Web Based Accommodation can be categorized module-wise. Basically all the information is managed by the software and in order to access the information one has to produce one's identity by entering the user-id and password. Every user has their own domain of access beyond which the access is dynamically refrained rather denied.
Output:
The major outputs of the system are tables and reports. Tables are created dynamically to meet the requirements on demand. Reports, as it is obvious, carry the gist of the whole information that flows across the institution. This application must be able to produce output at different modules for different inputs.

Performance Requirements:
Performance is measured in terms of reports generated weekly and monthly.

Intended Audience And Reading Suggestions:
The document is prepared keeping is view of the academic constructs of my Bachelor's Degree / Master's Degree from university as partial fulfillment of my academic purpose the document specifies the general procedure that that has been followed by me, while the system was studied and developed. The general document was provided by the industry as a reference guide to understand my responsibilities in developing the system, with respect to the requirements that have been pin pointed to get the exact structure of the system as stated by the actual client.
The system as stated by my project leader the actual standards of the specification were desired by conducting a series of interviews and questionnaires. The collected information was organized to form the specification document and then was modeled to suite the standards of the system as intended.

Document Conventions:

The overall documents for this project use the recognized modeling standards at the software industries level.

- ER-Modeling to concentrate on the relational states existing upon the system with respect to Cardinality.
- The Physical dispense, which state the overall data search for the relational key whereas a transaction is implemented on the wear entities.
- Unified modeling language concepts to give a generalized blue print for the overall system.
- The standards of flow charts at the required states that are the functionality of the operations need more concentration.

Scope of the Development Project:

Database Tier:

The concentration is applied by adopting the Oracle 8i Enterprise versions. SQL is taken as the standard query language. The overall business rules are designed by using the power of PL/SQL components like stored procedures stored functions and database triggers.

User Tier:

The use interface is developed is a browses specific environment to have distributed architecture. The components are designed using HTML standards and Java server pages power the dynamic of the page design.

Data Base Connectivity Tier:

The communication architecture is designed by concentrated on the standards of servlets and JSP's. The database connectivity is established using the Java Database connectivity.

Role Of Oracle In Database:

ORACLE 8i is one of the many database services that plug into a client / server model. It works efficiently to manage resources, a database information, among the multiple clients requesting & sending.

Structured Query Language (SQL)

SQL is an inter-active language used to query the database and access data in database. SQL has the following features:

1. It is a unified language.
2. It is a common language for relational database
3. It is a non-procedural language.

Feasibility Report:

System Analysis Concentration:

- Before planning a replacing for a new system it is essential to have through knowledge about the existing system along with estimation of how lost computes can be used to make its operations more effective.
- System analysis is the process of collecting and interpreting facts, disposing problem and use the information about the existing system, which is also called as system study.
- System analysis is about understanding situation but not solving the problem.
- System analysis is performed to determine whether a not it is feasible to design and information system laved on the policies and plans of an organization. To determine the user requirements and to eliminate the weakness of the present system a few general requirements are concerned.

GENERAL REQUIREMENTS:

- The new system should be cost effective
- To improve productivity and service and service.
- To enhance user interface.
- To improve information presentation and durability.
- To upgrade systems reliability, availability and flexibility.
- To address human factors for better and uses acceptance.

PROBLEM IN THE CURRENT SYSTEM:

The present system is presently is an undeveloped form and the manual process of the overall system is too clumsy and complicated. The clients in the real time consultancy system can be too thick and may need many resources to be used upon the system. If the system is developed, in a distributed over interface with centralized database is the only solution.

TECHINICAL FEASIBILITY:

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point in time, not too many detailed design of the system, making it difficult to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis.

Understand the different technologies involved in the proposed system:

Before commencing the project, we have to be very clear about what are the technologies that are to be required for the development of the new system.

Find out whether the organization currently possesses the required technologies:

Is the required technology available with the organization?

If so is the capacity sufficient?

For instance –

"Will the current printer be able to handle the new reports and forms required for the new system?"

URL that has been allocated for him and can make a wide variety of choices for the accommodations that are available in the central repository of the existing database. The User can have the information of all the units that are practically registered along with the facilities that are available with respect to every unit. The guest is given free hands to register his information as per the practical requirements of the site, such that he can be tracked and provided services at the demand that is raised by him.

The Guest once registered can make multiple visits upon the site as per his requirements and enquire for the unit's availability and the status of his previous bookings, if any.

The Administrators can make easy accessibility of the system from virtually anywhere in this world, which facilitates the scope of global or remote administration possible. All the consistent transactions are facilitated by the administrators only with proper authorization and authentication.

Cost Based:

If the physical system is established through a manual process. There is much need of stationary that has to be managed and maintained as files, the overall system once implemented as a intranet based web application not only saves the time but also eliminates the latency that can exist within the system, and saves the cost of stationary that is an unforeseen overhead within the system.

The base administrative staff at the level of the strategic decision-making is greatly relieved for the extensive data search. The administrators are greatly benefited by this system, as they need not collect all that information that is accessible by only selects that information which is more important for them. The administrative standards of the system become more economical as there is no need of stationary exchange within the organization. As the information governed upon the system maintains statistics related to information that is collected, the overall system can be used for forecasting analysis in the research process.

In the manual process of the Web Based Accommodation Upholding and Maintenance System, the information search and storage needs extra manpower assignment, which potentially costs the organization in the perennial investment of funds for the sake of the salaries. The information interrelations among different areas of the system should be handled carefully else the latency upon the overall process of the system increases leading to inconvenience.

Functional Requirements:

Inputs: The major inputs for Integration of Web based Accommodation Upholding Maintenance System can be categorized module -wise. Basically all the information is managed by the software and in order to access the information one has to produce one's identity by entering the user-id and password. Every user has their own domain of access beyond which the access is dynamically refrained rather denied.

Output: The major outputs of the system are tables and reports. Tables are created dynamically to meet the requirements on demand. Reports, as it is obvious, carry the gist of the whole information that flows across the institution.

This application must be able to produce output at different modules for different inputs.

Performance Requirements:

Performance is measured in terms of reports generated weekly and monthly.

Intended Audience And Reading Suggestions

The document is prepared keeping is view of the academic constructs of my Bachelor's Degree / Master' Degree from university as partial fulfillment of my academic purpose the document specifies the general procedure that that has been followed by me, while the system was studied and developed. The general document was provided by the industry as a reference guide to understand my responsibilities in developing the system, with respect to the requirements that have been pin pointed to get the exact structure of the system as stated by the actual client.

The system as stated by my project leader the actual standards of the specification were desired by conducting a series of interviews and questionnaires. The collected information was organized to form the specification document and then was modeled to suite the standards of the system as intended.

Document Conventions:

The overall documents for this project use the recognized modeling standards at the software industries level.

ER-Modeling to concentrate on the relational states existing upon the system with respect to Cardinality.

The Physical dispense, which state the overall data search for the relational key whereas a transaction is implemented on the wear entities.

Unified modeling language concepts to give a generalized blue print for the overall system.

The standards of flow charts at the required states that are the functionality of the operations need more concentration.

Scope of the Development Project:

Database Tier: The concentration is applied by adopting the Oracle 9i Enterprise versions. SQL is taken as the standard query language. The overall business rules are designed by using the power of PL/SQL components like stored procedures stored functions and database triggers.

User Tier: The use interface is developed is a browses specific environment to have distributed architecture. The components are designed using HTML standards and Java server pages power the dynamic of the page design.

Data Base Connectivity Tier: The communication architecture is designed by concentrated on the standards of servlets and

JSP's. The database connectivity is established using the Java Database connectivity.

SYSTEM TESTING AND IMPLEMENTATION

INTRODUCTION

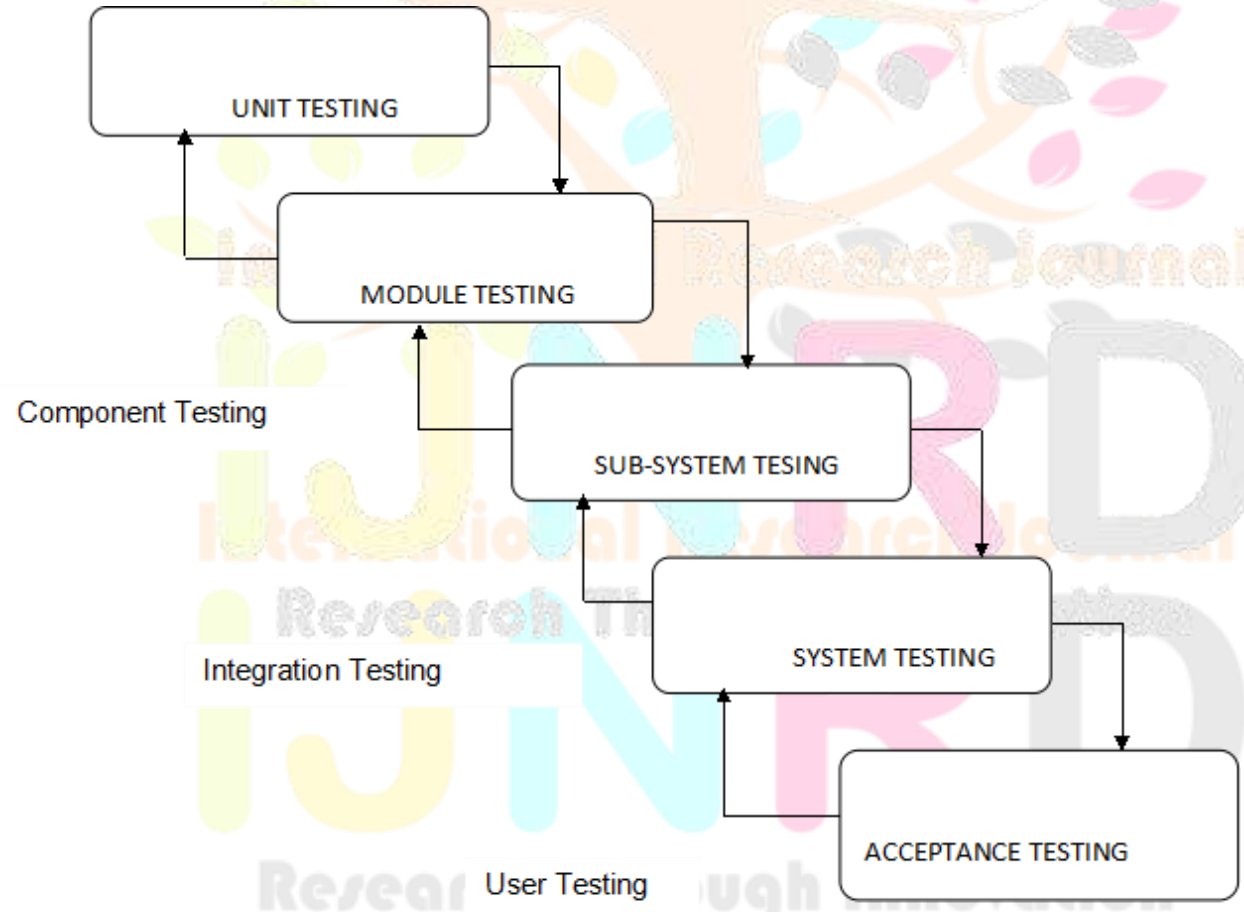
Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

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 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.



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

Established technique of flow graph with Cyclomatic 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 graph.

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

V(G)=E-N+2 or

V(G)=P+1 or

V(G)=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 generate 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 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 loops test the inner most loop first and then work outwards.

For concatenated loops the values of dependent loops were set with the help of connected loop.

Unstructured loops were resolved into nested loops or concatenated loops and tested as above.

Each unit has been separately tested by the development team itself and all the input have been validated.

OPERATIONAL FEASIBILITY:

Proposed projects are beneficial only if they can be turned into information systems that will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? Here are questions that will help test the operational feasibility of a project:

Is there sufficient support for the project from management from users? If the current system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.

Are the current business methods acceptable to the user? If they are not, Users may welcome a change that will bring about a more operational and useful systems.

Have the user been involved in the planning and development of the project?

Early involvement reduces the chances of resistance to the system and in

General and increases the likelihood of successful project.

Since the proposed system was to help reduce the hardships encountered. In the existing manual system, the new system was considered to be operational feasible.

ECONOMIC FEASIBILITY:

Economic feasibility attempts 2 weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system.

A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation. These could include increased customer satisfaction, improvement in product quality better decision making timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information, better employee morale.

a) **Technical Description**

Databases: The total number of databases that were identified to build the system is 10.The major part of the Databases is categorized as Administrative components and the user components. The administrative components are useful is managing the actual master data that may; be necessary to maintain the consistency of the system. The administrative databases are purely used for the internal organizational needs and necessities. The user components are designed to handle the transitional state that arise upon the system whenever the general client makes a visit onto the system for the sake of the report based information.

The user components are scheduled to accept parametrical information for the user as per the systems necessity.

GUI's

For the flexibility of the user, the interface has been developed in graphical user interface mode. The normal interface is applied through browser.

The GUI's at the top level has been categorized as:

1) Administrative user interface
2) Customer or general user interface

The administrative user interface concentrates on the consistent information that is practically, pact of the organizational activities and which needs proper authentication for the data collection. The interfaces help the visitors with all the transactional states like Data insertion, Data deletion and Data updating with the data search capabilities.

The general user interface helps the users upon the system in transactions through the required services that are provided upon the system. The general user interface also helps the ordinary user is managing their own information in a customized manner as per their flexibilities.

Time Based:

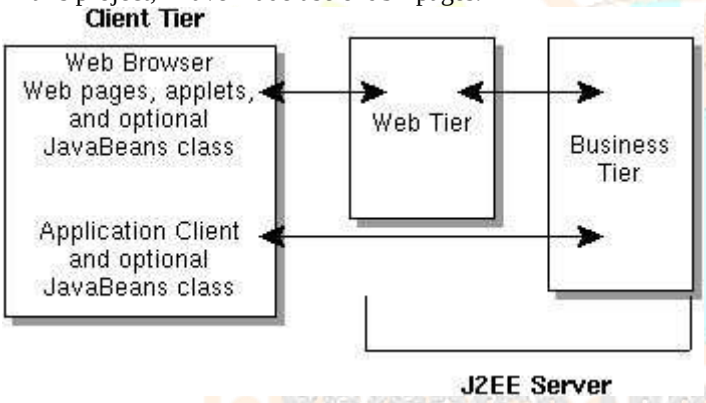
In this world of busy schedule with which the industrial professionals are getting through this kind of system is a boon for the kind of information they can readily access at the tip of their fingers. The Tourist, who intends to make a visit to the specific choice of his place, need not enquire about the details taking the entire pin physically. The Tourist can just get himself connected to the site at his own desk with the respective

Components of J2EE FRAMEWORK:

J2EE applications consist of Client and Web tier.

A J2EE component is one of the self-contained functional units inside a J2EE application. It functions independently, bundled and/or when interacting with various J2EE components written in the Java language and executed like any other Java program. Combining respective classes and files facilitates communication between J2EE components in the same J2EE application. Client tier components, such as application clients and applets, run on a client machine. Web and business tier components, such as Java servlets and Java server pages (JSP), run on a J2EE server.

In this project, I have made use of JSP pages.



J2EE web components can be either JSP pages or servlets. Servlets are Java programming language classes that dynamically process requests and construct responses. JSP pages are text-based documents that contain static content and snippets of Java programming language code to generate dynamic content. When a JSP page loads, a background servlet executes the code snippets and returns a response.

Static HTML pages and applets are bundled with web components during application assembly, but are not considered web components by the J2EE specification. Server-side utility classes can also be bundled with web components, and like HTML pages, are not considered web components.

J2EE Architecture:

The component-based and platform-independent J2EE architecture makes J2EE applications easy to write because business logic is organized into reusable components and the J2EE server provides underlying services in the form of a container for every component type.

Containers and Services:

Components are installed in their containers during deployment and are the interface between a component and the low-level platform-specific functionality that supports the component. Before a web, enterprise bean, or application client component can be executed, it must be assembled into a J2EE application and deployed into its container.

The assembly process involves specifying container settings for each component in the J2EE application and for the J2EE application itself. Container provides the underlying support provided by the J2EE Server such as security, transaction management, Java Naming and Directory Interface (JNDI) lookups, and remote connectivity.

Here are some of the advantages:

- The J2EE security model lets you configure a web component or enterprise bean so system resources are accessed only by authorized users.
- The J2EE transaction model lets you specify relationships among methods that make up a single transaction so all methods in one transaction are treated as a single unit.

- JNDI lookup services provide a unified interface to multiple naming and directory services in the enterprise so application components can access naming and directory services.
- The J2EE remote connectivity model manages low-level communications between clients and enterprise beans. After an enterprise bean is created, a client invokes methods on it as if it were in the same virtual machine.

Types of J2EE Containers:
An Enterprise JavaBeans (EJB) container manages the execution of all enterprise beans for one J2EE application. Enterprise beans and their container run on the J2EE server.

- A web container manages the execution of all JSP page and servlet components for one J2EE application. Web components and their container run on the J2EE server.
- An application client container manages the execution of all application client components for one J2EE application. Application clients and their container run on the client machine.
- An applet container is the web browser and Java Plug-in combination running on the client machine.

SYSTEM DESIGN

INPUT DESIGN:
Input design is the process of getting user originated input and it use it for the progressing of further operations in the project. The goal of designing input data is to make as easy logical error free as possible. It provides interactive environment between user and the system. Inaccurate input data are the most common cause of errors in data processing.

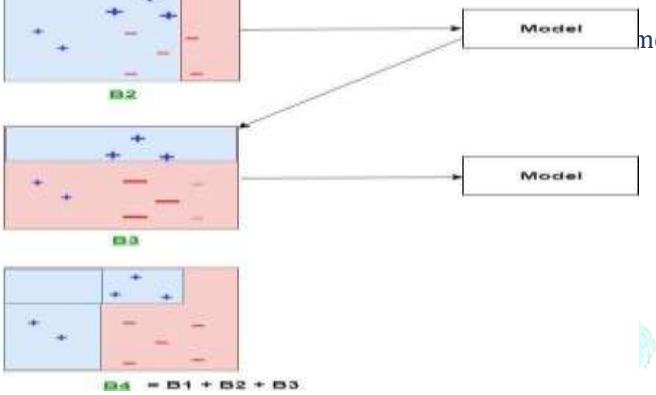
OUTPUT DESIGN:
While designing the output of the system the following factors should be considered.

- Determine what the information to present.
- Decide on the mode of output like to display, print, the information and select the Output medium.
- Arrange the presentation of information in an acceptable format. Computerized output is the most important as it's the direct source of information to the user. Efficient, Intelligible output design should improve the system's relationship with user and helps in decision making.

Output design generally refers to the results generated by the system. Many end users evaluate the usefulness of the application on the basis of the output. Efficient software must be able to produce efficient and effective results.
These activities required specific decisions such as whether to use Preprinted forms when preparing reports and documents, how many lines to print a page or whether to use the graphic and color. The output design is specified on a layout forms.
In the Production Planning application, the output can be view in the two ways. The output can be viewed within the application or as separate reports generated in the form of PDF files. These PDF files are saved in a separate location in the system.

ALGORITHM:

1. Initialise the dataset and assign equal weight to each of the data point.
2. Provide this as input to the model and identify the wrongly classified data points.
3. Increase the weight of the wrongly classified data points.
4. if (got required results)
   Goto step 5
   else
   Goto step 2
5. End



EXPLANATION:

The above diagram explains the AdaBoost algorithm in a very simple way. Let's try to understand it in a stepwise process:

- **B1** consists of 10 data points which consist of two types namely plus(+) and minus(-) and 5 of which are plus(+) and the other 5 are minus(-) and each one has been assigned equal weight initially. The first model tries to classify the data points and generates a vertical separator line but it wrongly classifies 3 plus(+) as minus(-).
- **B2** consists of the 10 data points from the previous model in which the 3 wrongly classified plus(+) are weighted more so that the current model tries more to classify these pluses(+) correctly. This model generates a vertical separator line that correctly classifies the previously wrongly classified pluses(+) but in this attempt, it wrongly classifies three minuses(-).
- **B3** consists of the 10 data points from the previous model in which the 3 wrongly classified minus(-) are weighted more so that the current model tries more to classify these minuses(-) correctly. This model generates a horizontal separator line that correctly classifies the previously wrongly classified minuses(-).
- **B4** combines together B1, B2, and B3 in order to build a strong prediction model which is much better than any individual model used.

FAST TEXT ALGORITHM:

What is FastText?

FastText is an open-source, free library from Facebook AI Research(FAIR) for learning word embeddings and word classifications. This model allows creating unsupervised learning or supervised learning algorithm for obtaining vector representations for words. It also evaluates these models. FastText supports both CBOW and Skip-gram models.

USES OF FASTTEXT:

1. It is used for finding semantic similarities
2. It can also be used for text classification(ex: spam filtering).
3. It can train large datasets in minutes.

WORKING OF FASTTEXT:

FastText is very fast in training word vector models. You can train about 1 billion words in less than 10 minutes. The models built through deep neural networks can be slow to train and test. These methods use a linear classifier to train the model.

Linear classifier: In this text and labels are represented as vectors. We find vector representations such that text and it's associated labels have similar vectors. In simple words, the vector corresponding to the text is closer to its corresponding label. To find the probability score of a correct label given it's associated text we use the softmax function:

$$softmax(w_{travel}) = \frac{e^{w_{car}^T * w_{travel}}}{\sum_{l \in labels} e^{w_{car}^T * w_l}}$$

Here travel is the label and car is the text associated to it.

To maximize this probability of the correct label we can use the *Gradient Descent algorithm*.

This is quite computationally expensive because for every piece of text not only we have to get the score associated with its correct label but we need to get the score for every other label in the training set. This limits the use of these models on very large datasets.

CONCLUSION:

For behavioral models in online payment fraud detection, we propose an effective data enhancement scheme by modeling co-occurrence relationships of transactional attributes. Accordingly, we design customized cooccurrence relation networks, and introduce the technique of heterogeneous network embedding to represent online transaction data for different types of behavioral models, e.g., the individual-level and population-level models. The methods are validated by the implementation on a real-world dataset. They outperform the stateof-the-art classifiers with lightweight feature engineering methods. Therefore, our methods can also serve as a feasible paradigm of automatic feature engineering. There are some interesting issues left to study.An interesting future work is to extend the data enhancement scheme into other types of behavioral models, e.g., the group-level models and generalized-agent-based models, except the population-level and individual-level models studied in this work. It would be interesting to investigate the dedicated enhancement schemes for more advanced individual-level models, since the adopted naive individual-level model does not fully capture the advantages of the proposed data representation scheme based on the techniques of heterogeneous network embedding. It is anticipated to demonstrate the generality of the proposed method by applying it to different

real-life application scenarios.

FUTURE ENHANCEMENT:

- In Future Enhancement, in this Project we can include some additional features from banking side such as loan payment and some transaction Process etc.
- We can enhance the overall report monitoring and tracking the transaction to avoid corruption in future.

REFERENCES:

[1] S. Bashir, D. T. C. Lai, and O. A. Malik, ‘‘Proxy-ters based query obfuscation technique for private web search,’’ IEEE Access, vol. 10, pp. 17845–17863, 2022.
[2] MOHIB ULLAH 1,2, RAFIULLAH KHAN 1,2, MUHAMMAD INAM UL HAQ3, ATIF KHAN 4, WAEL ALOSAIMI 5, MUHAMMAD IRFAN UDDIN 6, AND ABDULLAH ALHARBI, Multi-Group ObScure Logging (MG-OSLo) A Privacy-Preserving Protocol for Private Web Search, VOLUME 9, 2021, May 10, 2021,
[3] JAVERIAH SALEEM , RAFIQUL ISLAM , (Senior Member, IEEE), AND MUHAMMAD ASHAD KABIR, The Anonymity of the Dark Web: A Survey, VOLUME 10, 2022
[4] JINGYUAN LIANG 1, CHANSU YU 1, (Senior Member, IEEE), KYOUNGWON SUH2, (Member, IEEE), AND HYOIL HAN2, Tail Time Defense Against Website Fingerprinting Attacks, VOLUME 10, 2022
[5] Rajorshi Biswas and Jie Wu Department of Computer and Information Sciences Temple University, Philadelphia, Optimal Filter Assignment Policy Against Distributed Denial-of-Service Attack, 1545-5971 (c) 2020 IEEE. Personal use is permitted.
[6] Gilad Cohen, Guillermo Sapiro, and Raja Giryes. Detecting adver- sarial samples using influence functions and nearest neighbors. In The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.
[7] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Mma training: Direct input space margin max-imization through adversarial training. In International Conference on Learning Representations, 2020.
[8] Luis Mu ˜noz-Gonz´alez, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C. Lupu, and Fabio Roli. To- wards poisoning of deep learning algorithms with back-gradient optimization. CoRR, abs/1708.08689, 2017.
[9] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Evading real-time person detectors by adversarial t-shirt. CoRR, abs/1910.11099, 2019.
[10] B. Wang and N. Z. Gong. Stealing hyperparameters in machine learning. In 2018 IEEE Symposium on Security and Privacy (SP), pages 36–52, 2018.