1. PROJECT PROFILE

1.1 PROJECT DEFINITION

WeaveCloud is an application which is launched because of need for a destination where the computer users could come together and help each by sharing data. Users who are registered on WeaveCloud can upload, download, email and share their data and make new friends while doing all of the above in form of online groups. The project consists of simple and user friendly user-interface, providing users with basic features.

PURPOSE

The main purpose of WeaveCloud is to provide a cloud to users. It's a portable application wherein users can upload and download their data.

In other words, users can take back-ups of their useful data. The data can be any form of file or documents. Moreover, the users can do all this in a few clicks of the mouse button i.e., selecting a file to download or upload. This all can be done from anywhere while sitting at home or office.

What is WeaveCloud all about?

"Data Whenever, Wherever"

WeaveCloud is a simple application that is platform independent and runs through a communication network in which a client system communicates with the server. The server computer is by default set as "www.weavebytes.com" .However, users can change their server address accordingly in the individual user settings present in desktop application. The client system connects to the server system to obtain the user account details and settings. A new user can also register himself through a simple registration form.

After successful login into his account a user is presented with a simple user interface. A tree structure is present, which shows the user, the hierarchy of his file structure and directories in his account. If a user wishes to upload a file, he needs to click on upload button. The user can choose the file from file browser dialog box and then his file will be added to the tree.

Similarly, a user can select a file from the tree and click download button to start downloading the file onto his computer. The user can also delete files, create new folders, and do similar file operations.

WeaveCloud also provides the users with feature of groups. Registered users can create new groups and also join existing groups. Groups are means of communication and interaction between individual users.The system has an administrator who keeps an eye on the overall functioning of the system.

1.2 SCOPE

Following are the features which are in the scope of the software:.

- Provide a user friendly environment to new as well as registered users so they can register and use the cloud services.
- To maintain and store individual user data, user details, account details.
- Provide excellent and easy to access means of communication medium between client systems and WeaveCloud server.
- Provide WeaveCloud service to enable its users to connect with one another through groups and emails.
- Provide users with the ability to change and maintain individuals user settings like themes, look, connection settings, storage directory and other individual user settings.
- Provide a simple user friendly tree structure of user directories so users can manage files easily.
- Provide users the feature to upload and download files.

OBJECTIVE

The summary of the objective of this software is to completely automate the process of:

- Providing a simple user friendly interface to users to manage files in their accounts.
- Providing feature of uploading and downloading files to and from user accounts on the cloud.
- Maintaining and storing individual user data, user details, account details.
- Providing users with individual settings for their accounts.
- Provide WeaveCloud as a service to enable its users to connect with one another through groups and emails.

What contribution would the Project Make?

This is an era of Information Technology where security and storage of data is the base of each and every thing. Storage of data, security and ease of access to data at the required time , matter a lot to every user. Today, every person wishes to save his data from any place and able to access the same data from other place, in other words portability. The project would help in effective and systematic storage and access to the individual user data from any place that user wishes to access from. Invalid access by any person will be caught at the time of login. Moreover, this project is platform independent. Users can access their data from a Linux-based as well as Windows-based operating system.

2. SYSTEM STUDY & PROBLEM FORMULATION

2.1 THE EXISTING SYSTEM

Client server model

File Transfer Protocol (FTP) is a standard network protocol used to copy a file from one host to another over a network, such as the Internet. FTP is built on a client-server architecture and utilizes separate control and data connections between the client and server.FTP users may authenticate themselves using a clear-text sign-in protocol but can connect anonymously if the server is configured to allow it.

Peer-to-Peer Model

Distributed architecture without the need for central coordination, with participants being at the same time both suppliers and consumers of resources (in contrast to the traditional client–server model).

Mainframe computer

Powerful computers used mainly by large organisations for critical applications, typically bulk data processing such as census, industry and consumer statistics, enterprise resource planning, and financial transaction processing

2.2 LIMITATIONS OF EXISTING SYSTEM

Some of the negative aspects of the existing system are as follows:

- Course of action is time consuming. Wherever users need to upload or download data, it may take longer duration.
- Existing Systems have a long registration process.
- Existing systems are only web based.
- Vulnerability to network attacks such as spoof attacks, brute force attacks, bounce attacks

- Not much user friendly and not much ease of use.
- Existing systems are require a large infrastructure and are costly.
- Not much secure as FTP and older protocols were not designed to encrypt its traffic. All transmissions are in clear text, and user names, passwords, commands and data can be easily read by anyone able to perform packet capture (sniffing) on the network.
- No users preferences and settings are available.
- Users cannot communicate with each other in any form such as emails, private messages or groups.
- Backup problems exist in system. You need to obtain and pay for disk space on an FTP server. This storage space is typically very limited or much more expensive per MB than our service.
- Files can be corrupted during FTP transfers without your knowledge.
- FTP connections are unreliable and often times out, so you need to resume the connection and figure out which files were not backed up.
- Vulnerable to Packet capture (sniffing).

Types of Network Attacks :

a) Bounce attacks:-To conform to the FTP protocol, the PORT command has the originating machine specify an arbitrary destination machine and port for the data connection. However, this behavior also means that an attacker can open a connection to a port of the attacker's choosing on a machine that may not be the originating client.

Making this connection to an arbitrary machine for unauthorized purposes is the FTP bounce attack. It can be done by port scanning, bypassing packet filters devices. nmap is a port scanner that can utilize an FTP bounce attack to scan other servers.

b) Spoof attacks:- In the context of network security, a spoofing attack is a situation in which one person or program successfully masquerades as another by falsifying data and thereby gaining an illegitimate advantage.

c) Brute force attacks:- In cryptography, a brute force attack is a strategy used to break the encryption of data. It involves traversing the search space of possible keys until the correct key is found. The selection of an appropriate key length depends on the practical feasibility of performing a brute force attack. By obfuscating the data to be encoded, brute force attacks are made less effective as it is more difficult to determine when one has succeeded in breaking the code.

d) Sniffing(Packet capture):- Sniffing application traffic simply means that the attacker is able to view network traffic and will try to steal credentials, confidential information, or other sensitive data. Anyone with physical access to the network, whether it is switched or via a hub, is likely able to sniff the traffic. Also, anyone with access to intermediate routers, firewalls, proxies, servers, or other networking gear may be able to see the traffic as well.

e) Port Stealing:- In this the attackers are able to steal a port, capture a batch of traffic, unsteal the port, and dump all the traffic to the real destination, resteal it---controlling the flapping semi-deliberately.

To overcome these, the proposed system has been suggested.

2.3 THE PROPOSED SYSTEM

The proposed system is WeaveCloud, which is a desktop based application. It is a simple, user friendly and easy to use application. It has greater accuracy and efficiency. It also takes only limited time for upload and download of user data. Moreover, it has attractive features like email, groups, themes and many more additional features.

It is very light weight and very easy to use with simple registration and login processes. It require just JRE (java runtime environment) installed on your operating system. As it is developed using java language so it is platform independent that follows WORE (Write Once Run Everywhere) principle. So it can be run on any operating system like windows, Linux-based OS, etc. The system or project is made using the concept of thread i.e. for every client a independent light weight thread is launched on server which require minimum CPU processing and memory.

The system includes a client application and a server module. The users are provided with client application on their desktops.

The user gets into the system using user name and a unique password. Each user has his own accessibility permission to accomplish his task flawlessly. The administrator on server also has a ID and password to get access to the system so that no unauthorized person is able to keep an eye on the working of the complete system.

IMPLEMENTATION

The system consists of a single server running on windows or Linux-based system and single/multiple clients connected through network / Ethernet cable and network configured. One system is used for running the server process and the other system is used for running the client process. Communication between the client and the server processes takes place using socket interface to the underlying communication protocols.

Server process:

This is the process which runs on the server machine i.e.WeaveCloud server. It is usually run first. It first registers its address with the system and listens at a specified port number for a request for connection from a client. As soon as it receives a request, it establishes a connection with the client process and initializes the thread for that client. It does not have a graphical display. It shows the current status for server module of application i.e.

- Listening to client
- Connecting to a client
- Waiting for a message/reply from connected client
- Sending a response to client in form of a packet
- Disconnect

Client process:

This process is started on the client or user machine**.** It consists of a simple and user-friendly Graphical User Interface (GUI).The client process initially sends a request for connection to the server process. Once the Server process accepts the request, a connection is established between client and the server. The Client processes the user request and forms a packet accordingly. It then sends the packet to server through the communication network. For every user action a new request is formed to send to the server in form of packet. The process in form of steps can be:

- Connecting to the server
- Sending a login/register request to the server
- Waiting for server response
- Form a new request in form of packet , according to user action

Storage:

The project does not include any database as such. This is because amount of data to be stored is not large and using a database would decrease speed and hence degrade the overall performance of application.

However, the data in this project is stored in form of files and hash tables. As a result, the application is light weight and has good performance. The data is stored in form of standard **INI files (.ini)** both on server and client side of application.

2.4 ADVANTAGES OF PROPOSED SYSTEM

- **Empowerment** of end-users of computing resources by putting the provisioning of those resources in their own control, as opposed to the control of a centralized IT service (for example)
- **Agility** improves with users' ability to re-use technological infrastructure resources.
- **Platform and location independence** enable users to access their data "and wherever"
- **Multi-tenancy** enables sharing of resources and costs across a large pool of users thus allowing for:
    - **Centralisation** of infrastructure in locations with lower costs (such as real estate, electricity, etc.)
    - **Peak-load capacity** increases (users need not engineer for highest possible load-levels)
    - **Better Utilisation and efficiency**

- **Reliability** is improved if multiple redundant sites are used, which makes well-designed cloud computing suitable for business continuity and disaster recovery.

- It is **desktop based** application which can be installed easily on any system
- **Open-Source** software.
- Fully works as an **online** system.
- **Multiple clients** can connect to server at same time.
- **Physical Space reduction:** The data is stored on server as backup.
- Easy access to the data.
- The new system is more users friendly and flexible.
- The users can do all this in a single click of the mouse button i.e. upload, download & basic cloud operations etc. can be done while anywhere.
- Simple and easy registration and login process.
- Reducing the time for basic file upload and download

DISADVANTAGES OF PROPOSED SYSTEM

Following are the disadvantages of the proposed system :-

- It requires an addition component (JRE) to run.
- Security has not yet been fully implemented such as secure logins and SSL

2.5 FEASIBILITY STUDY

Feasibility study is a test of a system proposal according to its workability impact on the organization, ability to meet user needs, and effective use of resources. The objective of a feasibility study is not to solve the problem but to acquire a sense of its scope. During the study, the problem definition is crystallized and aspects of the problem to be included in the system are determined. Consequently, costs and benefits are estimated with greater accuracy at this stage. The report produced at the end of the feasibility study contains suggestions and reasoned arguments to help management decide whether to commit further resources to the proposed project. The proposal summarizes what is known and what is going to be done.

Having gone through all measures of feasibility we report to the management to figure out if the objectives of the new system are met.

e.g. - Is the system within the budget allowed for it?

Will the organizations needs, be met by the new proposed system as

Originally envisaged?

If and when the objectives of the system are met and the new system is approved, then the more specific details in the proposal should be considered and approved.

There are various measures of feasibility that helps to decide whether a particular project is feasible or not. These measures include-

1) Economical Feasibility
2) Technical Feasibility
3) Behavioural Feasibility

A feasibility study is conducted to select the best system that meets performance requirement. This entails an identification description, an evaluation of candidate system and the selection of best system for the job. The system required performance is defined by a statement of constraints, the identification of specific system objective and a description of outputs.

Economical feasibility

It looks at the financial aspects of the project. It determines whether the management has enough resources and budget to invest in the proposed system and the estimated time for the recovery of cost incurred. It also determines whether it is worthwhile to invest the money in the proposed project. Economic feasibility is determined by the means of cost benefit analysis. The proposed system is economically feasible because the cost involved in purchasing the hardware and the software are within approachable. The software tool used is Java which is open source and free. The personal cost like salaries of employees hired are also nominal, because working in this system need not required a highly qualified professional. The operating-environment costs are marginal. The less time involved also helped in its economical feasibility. It was observed that the organization has already using computers for other purpose, so that there is no additional cost to be incurred for adding this system to its computers.

The backend required for storing other details is in form of files. The computers in the organization are highly sophisticated and don’t needs extra components to load the software. Hence the organization can implement the new system without any additional expenditure. Hence, it is economically feasible.

Benefits and Savings

- Cost of the maintenance of the proposed system is negligible.
- Money is saved as it is open source.
- Does not use any expensive and bulky database.
- Time is saved as all the work can be done by a simple mouse click.
- The proposed system is fully automated and hence easy to use.
- Since benefits out base the cost, hence our project is economically feasible.

Technical Feasibility

It is a measure of the practically of a specific technical solution and the availability of technical resources and expertise. This is concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may vary considerably, but might include:

- The facility to produce outputs in a given time.
- Response time under certain conditions.
- Ability to process a certain volume of transaction at a particular speed.
- Facility to communicate data to distant location.

In examining technical feasibility, configuration of the system is given more importance than the actual make of hardware. The configuration should give the complete picture about the system's requirements: How many workstations are required, how these units are interconnected so that they could operate and communicate smoothly. What speeds of input and out- put should be achieved at particular quality of printing. This can be used as a basis for the tender document against which dealers and manufacturers can later make their equipment bids. Specific hardware and software products can then be evaluated keeping in view with the logical needs. At the feasibility stage, it is desirable that two or three different configurations will be pursued that satisfy the key technical requirements but which represent different levels of ambition and cost. Investigation of these technical alternatives can be aided by approaching a range of suppliers for preliminary discussions. Out of all types of feasibility, technical feasibility generally is the most difficult to determine.

- The proposed system uses Java as front-end and files as back-end tool.
- Notepad++ was the text editor used for coding of application. It is a free and open-source text editor.
- Java API was used to refer to the java classes and as a help in coding.
- Adobe Photoshop and Paint are other tools used for photos and graphic editing including themes, icons etc.
- The above tools are easily available, easy to work with and widely used for developing commercial application.

Hardware used in this project are- Intel CORE i3 processor 2.26GHz, 4 GB RAM, 500 GB hard disk. This hardware was already available on the existing computer system. The software like JDK, Java API, Notepad++, Adobe Photoshop, Paint and operating system WINDOWS 7 used were already installed on the existing computer system. The application was also developed and tested on Ubuntu 10.04(Linux Based) system. So no additional hardware and software were required to purchase and it is technically feasible. The technical feasibility is in employing computers to the organization. The organization is equipped with enough computers so that it is easier for updating. Hence the organization has not technical difficulty in adding this system.

Tools Used :

1. JDK 1.7
2. Notepad++
3. Adobe Photoshop 7, CS3
4. Microsoft Paint

Behavioural Feasibility

The behavioural feasibility relates to the impact of proposed system on the user of the system that is how their feel about the proposed system. People are inherently resistance to change. An estimate should be made of how strong a reaction the user s is likely to have toward the development of a computerized system. It is common knowledge that computer installations have something to do with turnover, transfers, restraining, and changes in employee job status. Therefore, it is understandable that the introduction of the candidate system requires special effort to educate, sell, and train the user on new ways of conducting business.

How much training is required for the user? The proposed system is also behaviourally viable because of following reason.
Almost all of them generally work on the computers. Very little training is required for user to work on the proposed system, generally related to user commands & external interfaces. The system is capable of handling bulk of requests easily. Although, it is compatible to the changes, which might occur, needed for the better and fast services. The facilities provided are knowledge of handling the computer system.
People are inherent to change. In this type of feasibility check, we come to know if the newly developed system will be taken and accepted by the working force i.e. the people who will use it. The proposed system is very user friendly and easy to use for any user. It uses simple attractive buttons and easy to understand language.

- **Help Feature:** The proposed system has its own Help module which contains help topics regarding functioning of the software.
- **Tool Tips:** The system contains tips on each button and many labels, when the user points over them, which help the user to understand the functionality of corresponding button.

So the proposed system is behaviourally feasible.

Steps in Feasibility Analysis

Eight steps are involved in the feasibility analysis. They are:

i. Form a project team and appoint a project leader.

ii. Prepare system flowcharts.

iii. Enumerate potential proposed systems.

iv. Define and identify characteristics of proposed system.

v. Determine and evaluate performance and cost effectiveness of each proposed system.

vi. Weight system performance and cost data.

vii. Select the best-proposed system.

viii. Prepare and report final project directive to management

3 PROJECT PLAN

3.1 TEAM STRUCTURE

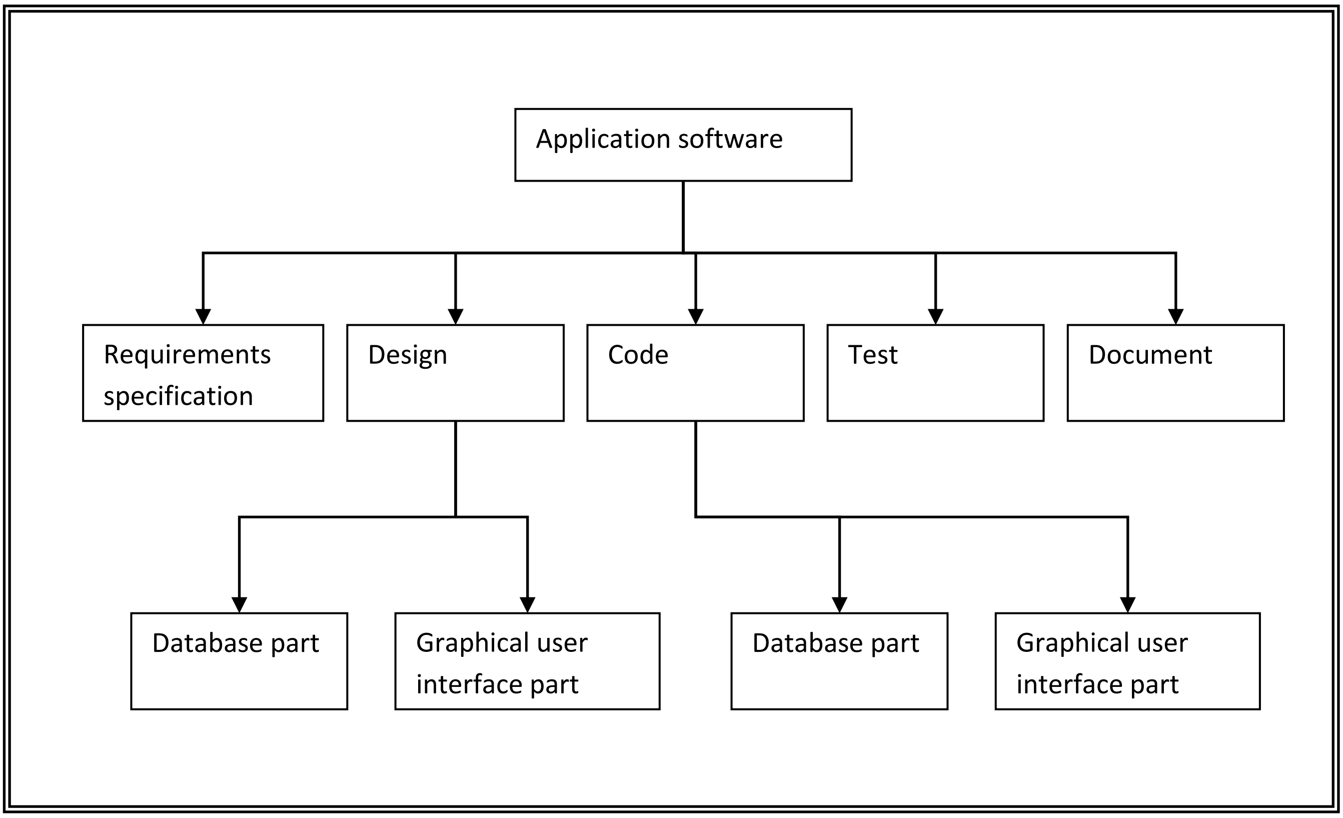
Introduction:

Software project managers take the overall responsibility of steering a project to success. This surely is a very hazy job description. But, it is very difficult to objectively describe the job responsibilities of a project manager. Often a team of people is assigned a project. But the team for this project consists of one project manager and one trainee. For team to work as a group and contribute most to the project, the people in the team have to be organized in some manner. This structure of Team has a direct impact on the product quality and project productivity. The structure of our team was democratic. Democratic team consists of ten or fewer. The structure allowed input from all the members, which led to better decisions in difficult situations.

WORK BREAKDOWN STRUCTURE

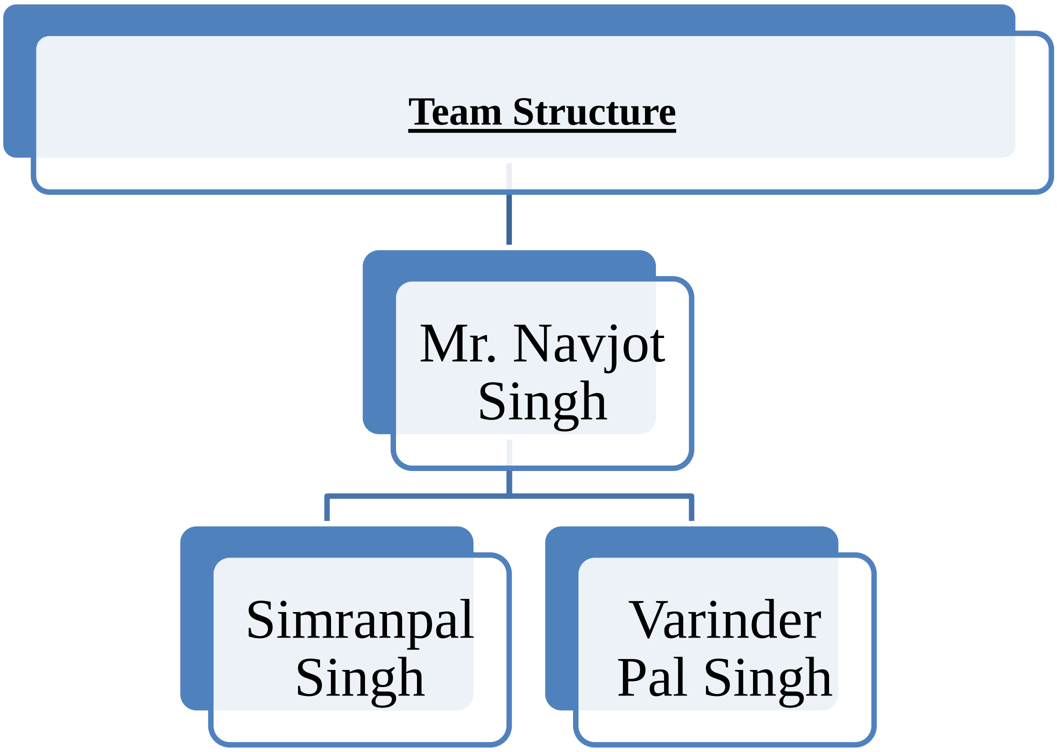
Work breakdown structure is used to decompose a given task set recursively into small activities. WBS provides a notation for representing the major tasks needed to be carried out in order to solve a problem. The root of the tree is labelled by the problem name. Each node of the tree is broken down into smaller activities that are made the children of the node. Each activity is recursively decomposed into smaller sub-activities until at the leaf level; the activities require approx. two weeks to develop. If a task is broken down into a large number of very small activities, these can be distributed to a large number of engineers. If the activity ordering permits, the solutions to these can be carried out independently. Thus, it becomes possible to develop the product faster.

The following figure represents the WBS of application software:


Application software
Requirements specification
Design
Code
Test
Document
Database part
Graphical user interface part
Database part
Graphical user interface part

Our team comprises of following members:

- Mr. Navjot Singh (The Project Manager)
- Mr. Simranpal Singh (The Trainee)
- Mr.Varinder Pal Singh (The Trainee)



3.2 DEVELOPMENT SCHEDULE

Software project management begins with a set of activities that are collectively called project planning. The objective of software project planning is to provide a framework that enables the manager to make reasonable estimates of resources, cost, and schedule. These estimates are made within a limited time frame at the beginning of a software project and should be updated regularly as the project progresses.

The work in developing the new system commented immediately with our first meeting with the management and users thereafter we were in continuous touch with the management.

Time Duration	
For study	**2 weeks**
Designing	**3 weeks**
For development	**8 weeks**
Testing	**2 weeks**
Documentation	**2 weeks**

Total time	**17 weeks**

Phases Involved in the development of the system

- Identification of needs and benefits
- Held Interactive sessions with the user to understand the requirements.
- Identification of needs, project constraints & Established project statements.
- Prepare the software requirement specifications.
- Structures Analysis & Design is done.
- Actual coding started.

3.3 PROGRAMMING LANGUAGE & DEVELOPMENT TOOLS

System Specification

The system specification is the final work produced by the system and requirements engineer. It serves as the foundation for hardware engineering, software engineering, database engineering and human engineering.

Hardware Specification

Programming language & Development Tools

HARDWARE

Processor	:	Intel CORE i3 processor 2.26GHz
HDD	:	500 GB
RAM	:	4 GB

SOFTWARE

Operating System	:	Microsoft Windows 7
Front End	:	JAVA
Database Used	:	Text File (.ini)

JAVA

Java: Introduction

Java is an object-oriented programming language with a built-in application programming interface (API) that can handle graphics and user interfaces and that can be used to create applications or applets. Because of its rich set of API's, similar to Windows, and its platform independence, Java can also be thought of as a platform in itself. Java also has standard libraries for doing mathematics.java does not have pointers.

The Java programming language is a high level language that can be characterized by all of the following buzzwords:

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

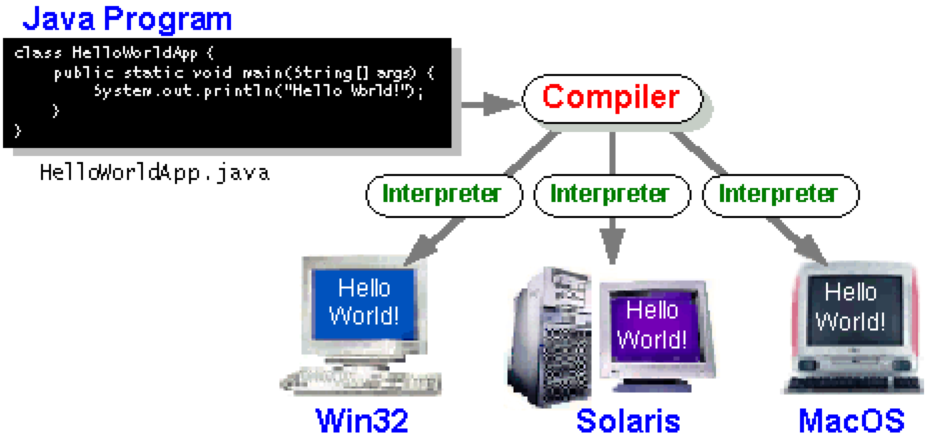
With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java 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. Platform Independent Java is a platform for application development. It means some combination of hardware and system software that will mostly run all the same software.

Java byte code is exactly the same on every platform.. Java programs that have been compiled into byte code still need an interpreter to execute them on any given platform. The interpreter reads the byte code and translates it into the native language of the host machine on the fly. Since the byte code is completely platform independent, only the interpreter and a few native libraries need to be ported to get Java to run on a new computer or operating system.

All these pieces, the javac compiler, the java interpreter, the Java programming language, and more are collectively referred to as Java.

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.



```
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Java: The 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.

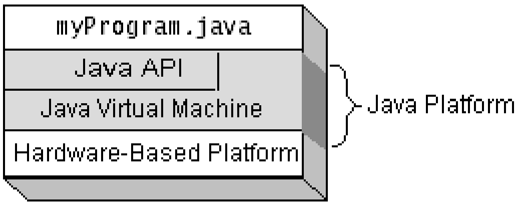
The Java platform has two components:

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

Java Virtual Machine is standardized hypothetical computer, which is emulated inside our computer by a program. It is base of 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 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.



What Can Java Technology Do?

The most common types of programs written in the Java programming language are applets and applications. If you've accessed 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.

AWT & Swing

AWT and Swing is a part of java extension which encompass a group of features to help people build graphical user interfaces (GUIs). Swing contains the following features:

- **The AWT Components**

  Include everything from buttons to split panes to tables.

- **Pluggable Look and Feel Support**

  It gives any program that uses Swing components a choice of looks and feels. For example, the same program can use either the Java look and feel or the Windows look and feel. We expect many more look-and-feel packages -- including some that use sound instead of a visual "look" -- to become available from various sources.

- **Accessibility API**

  It enables assistive technologies such as screen readers and Braille displays to get information from the user interface.

- **Java 2D^{TM} API (Java 2 Platform only)**

  It enables developers to easily incorporate high-quality 2D graphics, text, and images in applications and in applets.

- **Drag and Drop Support (Java 2 Platform only)**

Provides the ability to drag and drop between a Java application and a native application. The Swing API is available in two forms:

- As a core part of the Java 2 Platform, Standard Edition (including versions 1.2, 1.3, and 1.4)
- JFC 1.1 (for use with JDK 1.1)

Sun community recommends that to use the latest version of the Java 2 Platform. Not only will you be getting the latest bug fixes, but you'll get more features. (As we used JSDK1.5.0_01 version, which is the latest release of java development kit up to the time of written of this report.

Oops Concepts & Characteristics

Objects and Classes

Objects are the basic run time entities in an object-oriented system. They may represent a person, a place, a bank account, a table of data or any item that the program may handle. Each object contains data and code to manipulate the data. A class may be thought of as a 'data type' and an object as a 'variable' of that data type. In object-oriented programming, a class is a programming language construct that is used to group related instance variables and methods. A method, called a function in some languages, is a set of instructions that are specific to a class. Depending on the language, classes may support multiple inheritances or may require the use of interfaces to extend other classes. A class may indicate either specifically or abstractly what methods exist when the program is executed. The latter is known as an 'abstract class'.

Once a class has been defined, we can any number of objects belonging to that class.

Data abstraction

Data abstraction means hiding the object. A powerful way to manage abstraction is through the use of hierarchical classifications. This allows to layer the semantics of the complex systems, breaking them into more manageable pieces.

Polymorphism

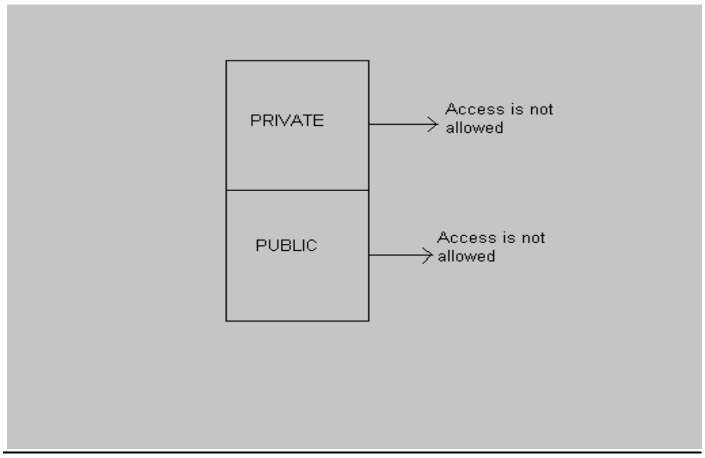
Polymorphism is a feature that allows one interface to be used for a general class of actions. Using operator or functions in different ways depending on what they are operating on, is called polymorphism. Polymorphism is of two types i.e. static and dynamic.

Inheritance

It is the most powerful feature of object oriented programming. It is the process of creating new classes called derived classes from the existing or base class. The derived class inherits all the properties of the base class but can add embellishments and refinements of its own.

Encapsulation

The concept of encapsulation and data hiding states that non member function should not be able to access an object’s private or protected data. Encapsulation is the mechanism that binds together code and data it manipulates, and keeps both safe from outside interference and misuse.



Packages

Packages and interfaces are two of the basic components of a Java program. In general, a Java source file can contain any (or all) of the following four internal parts:

- single package statement (optional)
- Any number of import statements (optional)
- A single public class declaration (required)
- Any number of classes private to the package (optional)

To create a package is quite easy: simply include a **package** command as the first statement in a Java source file. This is the general form of the **package** statement:

Package pkg;

Access Protection

Classes and packages are both means of encapsulating and containing the name space and scope of variables and methods. Packages act as containers for classes and other subordinate packages. Classes act as containers for data and code. The class is Java's smallest unit of abstraction. Because of the interplay between classes and packages, Java addresses four categories of visibility for class members:

- Subclasses in the same package
- Non-subclasses in the same package
- Subclasses in different packages
- Classes that are neither in the same package nor subclasses

The three access specifiers **private**, **public**, and **protected**, provide a variety of ways to produce the many levels of access required by these categories. Table sums up the interactions.

Class Member Access

	Private	**No modifier**	**Protected**	**Public**
Same class	Yes	Yes	Yes	Yes
Package non-subclass	No	Yes	Yes	Yes
Different package subclass	No	No	Yes	Yes

Exceptions

Exceptions are objects that describe any error caused by an external resource not being available or an internal processing problem. They are passed to exception handlers written by the programmer to enable graceful recovery. If the handler has not been written, the program will terminate with a display of the Exception class. There are many exception classes such as IOException and NumberFormatException etc.

Try/catch block

The code that might throw an exception is enclosed in the try block. One or more catch clauses can be provided to handle different exception types:

```
 try
  {
            // code that might throw exceptions
  }
  catch (Exception e)
 {
           //code to handle the exception
}
```

```
 finally()
{
        //code which always execute
}
```

The more specific exception type should be caught first. For instance, we have to order the catch blocks from specific to most general, otherwise a compile-time error occurs.

Characterstics of Java :

Object oriented

In object-oriented programs objects represent data. Objects have two sections, fields (instance variables) and methods. Fields tell you what an object is. Methods tell you what an object does. These fields and methods are closely tied to the object's real world characteristics and behavior. When a program is run messages are passed back and forth between objects. When an object receives a message it responds accordingly as defined by its methods.

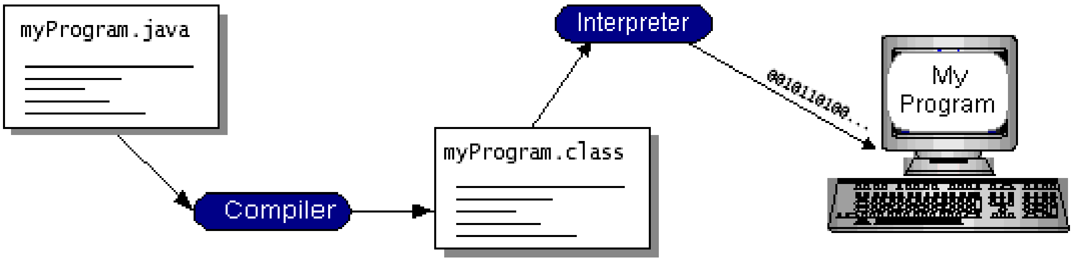
Robust

Java implements a robust exception handling mechanism to deal with both expected and unexpected errors.

Dynamic

Java programs carry with them substantial amounts of run-time type information that is used to verify and resolve accesses to objects at run-time. This makes it possible to dynamically link code in a safe and expedient manner.

The java programming language is unusual than other programming languages it first compiles and then interprets the program. Compile first translate the program into intermediate language called intermediate language called java byte code. Java byte code is platform independent code, which is further interpreted by the interpreter on the java platform. Interpreter parses and run each java byte code instruction on the computer. Compilation occurs only once, interpretation occurs each time when the program is executed.



Java byte code helps in making the program “write once, run anywhere”. The program can be compiled into byte code by any platform that is having the java compiler; the compiled java byte code program is ready to run on any machine having the java virtual machine (JVM). JVM is an interpreter for byte code.

Constructors

A constructor is a piece of code that runs when an instance of an object is created. A constructor is indicated by a method that:------

Has the same name as that of class

Has no return type specified

Interfaces

An interface defines the communication boundary between two entities, It generally refers to an abstraction that an entity provides of itself to the outside. This separates the methods of external communication from internal operation, and allows it to be internally modified without affecting the way outside entities interact with it, as well as provide multiple abstractions of itself.

Exception handling

Exception handling is a programming language construct or computer hardware mechanism designed to handle the occurrence of some condition that changes the normal flow of execution. The condition is called an exception. Exceptions are used only for signaling error (exceptional) conditions. For example, page fault, division by zero

Event Handling

Events – an events an object that describes a state change in a source. It can be generated as a consequence of a person interacting with the elements in a graphical user interface. Some of the activities that cause events to be generated are pressing a button, entering a character via keyboard, selecting an item in a list, and clicking the mouse.

Event Sources – a source is an object that generates an event. This occurs when the internal state of that object changes in some way. Sources may generate more than one type of event. A source must register listeners in order for the listeners to receive notifications about a specific event. Each type of event has its own registration method.

Event Listeners – a listener is an object that is notified when event occurs. It has two major requirements. First, it must have been registered with one or more sources to receive notifications about specific types of events. Second, it must implement methods to receive and process these notifications. Commonly used event listeners are ActionListener, MouseListener, WindowListener, MouseMotionListener, etc… .

Abstract Window Toolkit (AWT)

The Abstract Window Toolkit (AWT) is Java's original platform-independent windowing, graphics, and user-interface widget toolkit. The AWT is now part of the Java Foundation Classes (JFC) — the standard API for providing a graphical user interface (GUI) for a Java program.

Swing

Swing is a GUI toolkit for Java. It is one part of the Java Foundation Classes (JFC). Swing includes graphical user interface (GUI) widgets such as text boxes, buttons, split-panes, and tables. The Swing toolkit, shipped as part of the Java SE platform, provides a rich set of GUI components. But Swing offers much more functionality than a collection of standard widgets. This section takes a look at Swing's rich functionality.

Frame

A Frame is a top-level window with a title and a border. The size of the frame includes any area designated for the border. A frame, implemented as an instance of the JFrame class

Panels

The JPanel class provides general-purpose containers for lightweight components. By default, panels do not add colors to anything except their own background; however, you can easily add borders to them and otherwise customize their painting .

Labels

With the JLabel class, you can display unselect able text and images. If you need to create a component that displays a string, an image, or both, you can do so by using or extending JLabel.

Lists

A JList presents the user with a group of items, displayed in one or more columns, to choose from. Lists can have many items, so they are often put in scroll panes.

Combo Boxes

A JComboBox, which lets the user choose one of several choices, The default form is the uneditable combo box, which features a button and a drop-down list of values.

Buttons

The JButton class provides the functionality of a push button. JButton allows an icon, a string, or both to be associated with push button.

TextField and TextArea

A text field is a basic text control that lets the user enter a small amount of text. When the user indicates that text entry is complete (usually by pressing Enter), the text field fires an action event. If you need to obtain more than one line of input from the user, you should use a text area instead of TextField.

Advantages of the language (JAVA):-

The key that allows Java to solve both the security and the portability problems just described is that the output of a Java compiler is not executable code. Rather, it is bytecode. Bytecode is a highly optimized set of instructions designed to be executed by the Java run-time system, which is called the Java Virtual Machine (JVM). That is, in its standard form, the JVM is an interpreter for bytecode.

Simple

Java has the only required functionality needed to implement its rich features set. It doesn't add lots code and unnecessary features

Object oriented

Almost everything in java is either a class, or a method or an object. Only the most basic primitive operation and data types (int, for, while, etc) are at a sub-object level.

Platform Independent

Java programs are compiled to a byte code format that can be read and run by interpreters on many platforms including windows 95/NT and Solaris 2.3 and later.

Safe and secure

Java code can be executed in an environment that prohibits it form introduction viruses, deleting and modifying files, or otherwise performing data destroying and computer crashing operations.

Speed

Java can be compiled on the fly with a just-in-time (JIT) to code that rivals C++ in speed.

Multi-Threaded

Java is inherently multi threaded. A single java program has many different things processing independently and continuously.

Objects are classified in java using a class to encapsulate the data and behavior of the object. Like other object-oriented programming languages, the data of the object is hidden from rest of the system, and made available through the services of the class,

Classes are defined in java using the java keyword.

The attributes (data) of the class are written with the keyword private. This means the data is not accessible directly through the class. Instead, the methods of the class are used to access them. This ensures the integrity of the class and is one of the major strengths of object oriented programming.

The methods of the class define the behavior- the things the class can do. Users of the class may invoke the methods, so therefore have public access.

API

An **application programming interface** (**API**) is a particular set of rules and specifications that a software program can follow to access and make use of the services and resources provided by another particular software program that implements that API. It serves as an interface between different software programs and facilitates their interaction, similar to the way the user interface facilitates interaction between humans and computers.

API and Protocols

An API can also be an implementation of a protocol. In general the difference between an API and a protocol is that the protocol defines a standard way to exchange requests and responses based on a common transport, while an API provides a library to be used directly: hence there can be no *transport* (no information physically transferred from some remote machine), but rather only simple information exchange via *function calls* (local to the machine where the elaboration takes place).When an API implements a protocol it can be based on proxy methods for remote invocations that underneath rely on the communication protocol. The role of the API can be exactly to hide the detail of the transport protocol.Protocols are usually shared between different technologies (system based on given computer programming languages in a given operating system) and usually allow the different technologies to exchange information, acting as an abstraction/mediation level between the two *worlds*. While APIs are specific to a given technology: hence the APIs of a given language cannot be used in other languages, unless the function calls are wrapped with specific adaptation libraries.

Use of API's to share content

- Photos can be shared from sites like Flickr and Photobucket to social network sites like Facebook and MySpace.
- Content can be embedded, e.g. embedding a presentation from SlideShare on a LinkedIn profile.
- Content can be dynamically posted. Sharing live comments made on Twitter with a Facebook account, for example, is enabled by their APIs.
- Video content can be embedded on sites which are served by another host.

4 STRUCTURED ANALYSIS AND STRUCTURED DESIGN

Software project management begins with a set of activities that are collectively called project planning. The objective of software project planning is to provide a framework that enables the manager to make reasonable estimates of resources, cost, and schedule. These estimates are made within a limited time frame at the beginning of a software project and should be updated regularly as the project progresses.

STRUCTURED DESIGN

The most creative and challenging phase of the system development life cycle is system design. The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the first step in moving from the problem domain to the solution domain, in other words with what is needed, design takes us toward now to satisfy the need.

The design of a system is perhaps the most critical factor affecting the quality of the software. It has a major impact on the later phase, particularly testing and maintenance the design activity is often divided into two separate phases system design and detailed design.

In system design the focus is on identifying the modules that should be in the system , the specification of these modules and how they interact with each other to produce the desired result at the end of system design all the major data structures , file formats and major modules in the system and there specification are decided.

Design Principle:

Basic design principles enable the software engineer to navigate the design process. Principles for software design, which have been adapted and extended, are given in the following list:

- The design process should not suffer from “tunnel vision”. A good designer should consider alternative approaches and judging each based on the requirements of the problem.
- The design should be traceable to the analysis model. It is necessary to have a means for tracking how requirements have been satisfied by the design model.
- The design should not reinvent the wheel. Systems are constructed using a set of design patterns, many of which have likely been encountered before. Design time should be invested in representing truly new ideas and integrating those patterns that already exist.

- The design should “minimize the intellectual distance” between the software and the problem as it exists in the real world.
- The design should exhibit uniformity and integration. Rules of style and format should be defined for a design team before design work begins.
- The design should be structured to accommodate change.

4.1 DATA FLOW DIAGRAM

Data Flow Diagramming is a means of representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data processes, and data sources/destination.

The data flow diagram is analogous to a road map. It is a network model of all possibilities with different detail shown on different hierarchical levels. This processes of representing different details level is called “leveling” or “partitioning” by some data flow diagram advocates. Like a road map, there is no starting point or stop point, no time or timing, or steps to get somewhere. We just know that the data path must exist because at some point it will be needed. A road map shows all existing or planned roads because the road is needed.Details that is not shown on the different levels of the data flow diagram such as volumes, timing, frequency, etc. is shown on supplementary diagrams or in the data dictionary. For example, data store contents may be shown in the data dictionary.

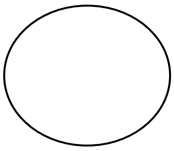
Data Flow Diagram (DFD) uses a number of symbols to represent the systems. Data Flow Diagram also known as ‘Bubble Chart’ is used to clarify system requirements and identifying the major transformations that will become programs in system design. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the level of details.

Terms used in DFD

- **Process**

A process transforms data values. The lowest level processes are pure functions without side effects. An entire data flow graphics high level process.

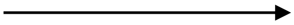
Graphical Representation:



- **Data flows**

A data flow connects the output of an object or process to input of another object or process. It represents the intermediate data value within a computation. It is represented by an arrow and labeled with a description of data, usually its name or type.

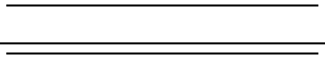
Graphical Representation:



- **Data store**

A data store is a passive object with in a data flow diagram that stores data for later acces

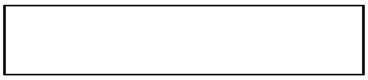
Graphical Representation:



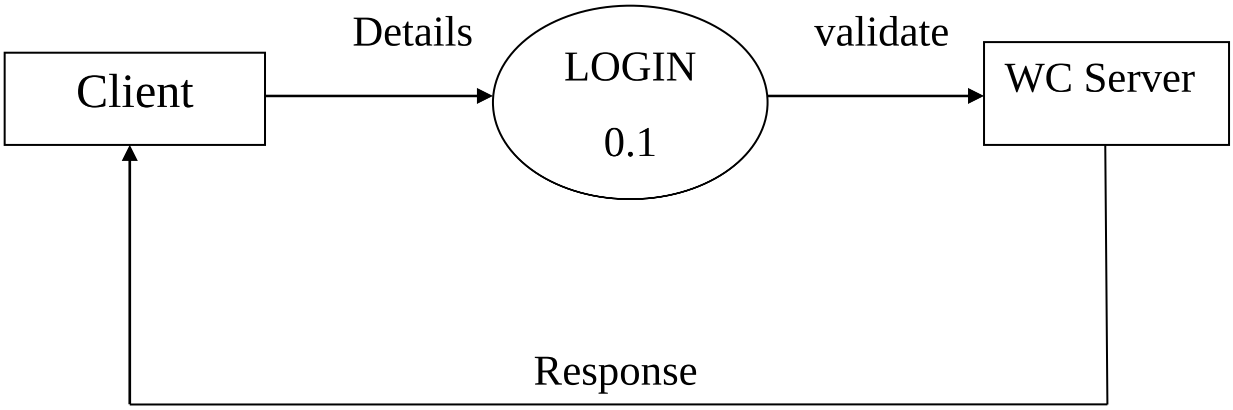
- **External Entity**

A rectangle represents an external entity such as a User,Admin,Viewer.

Graphical Representation:



<u>Data Flow Diagrams (DFD's)</u> - :



<u>Fig. Shows Context Level DFD for WeaveCloud</u>

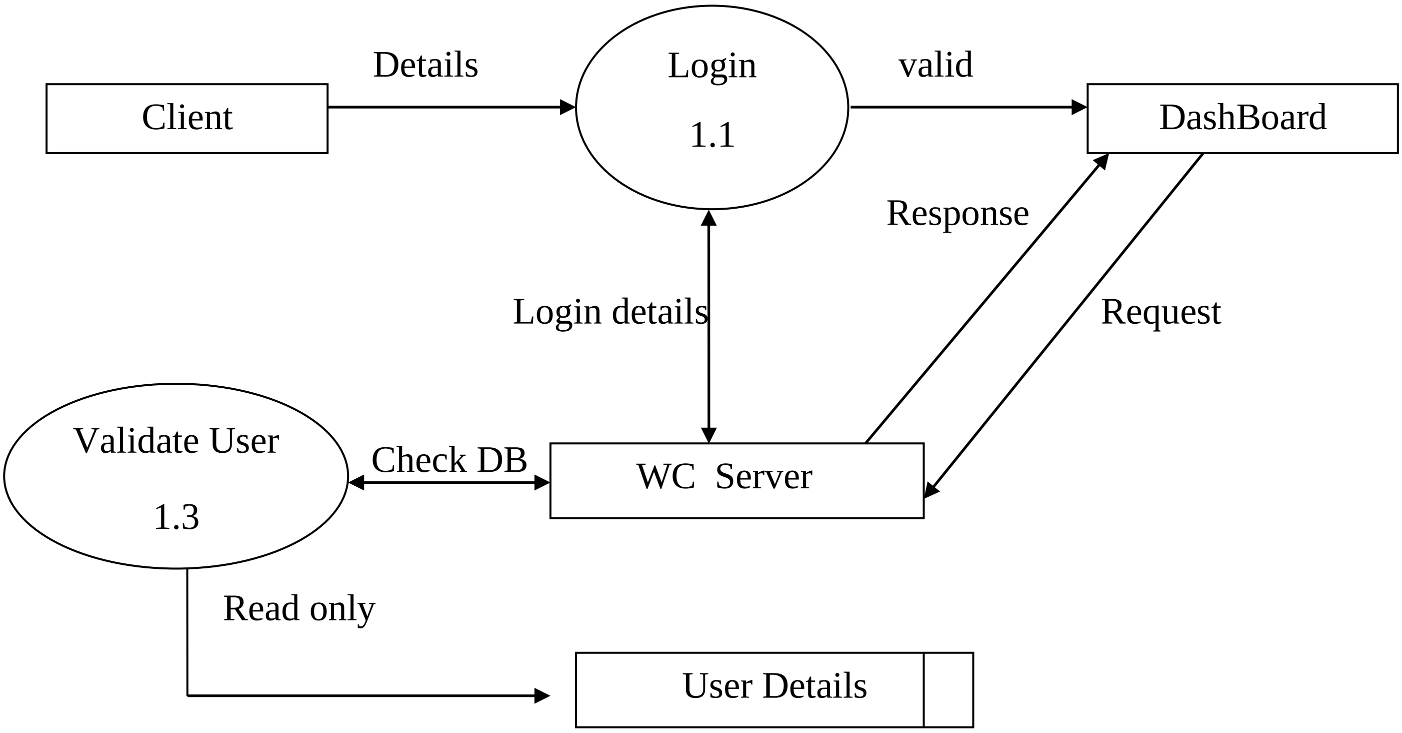


Fig. Shows Level 1 DFD for Login into WeaveCloud

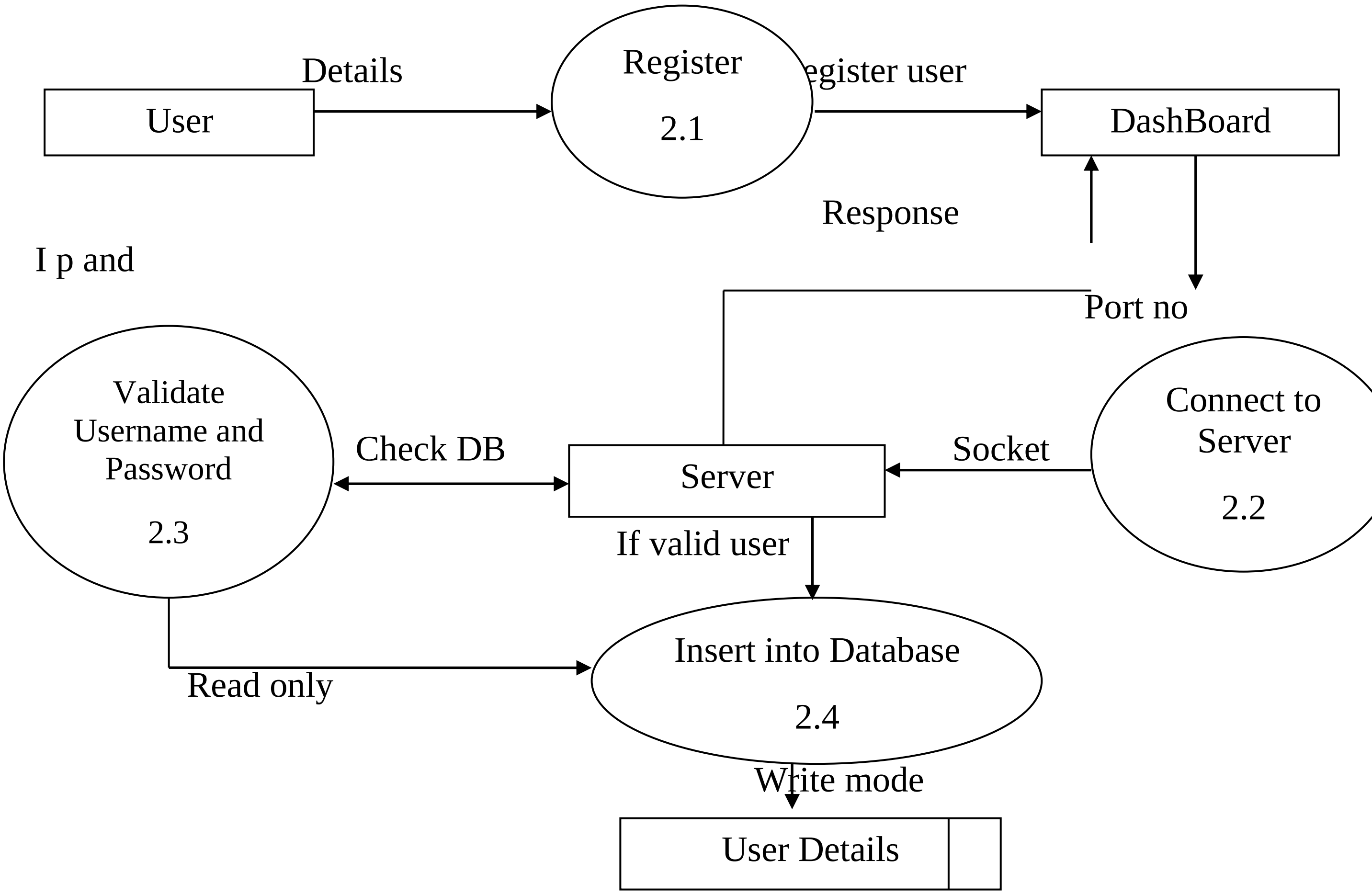


Fig. Shows Level 1 DFD for Registration into WeaveCloud

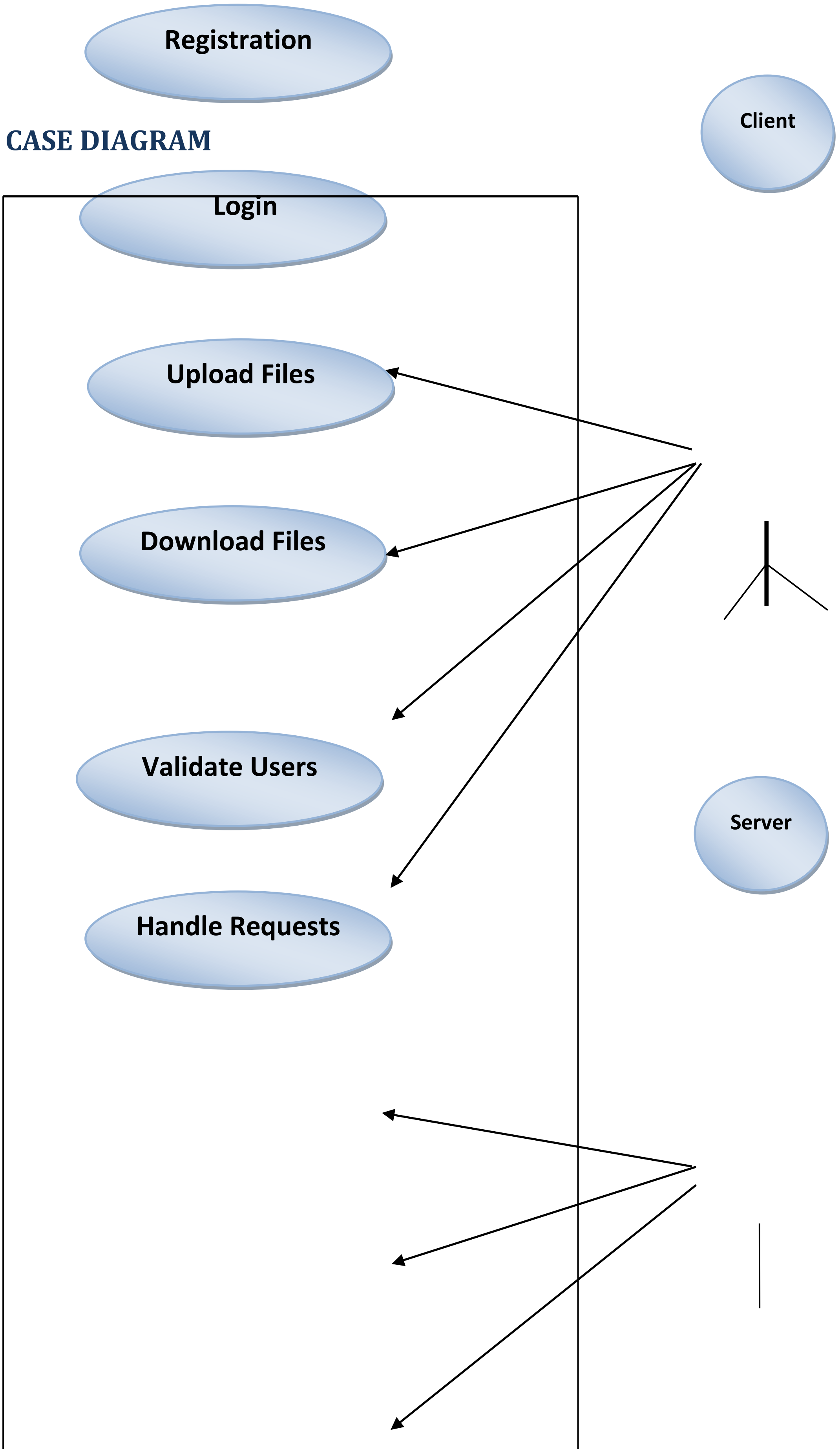
DATA DICTIONARY

Data dictionary is a structured repository of data about data. It is a set of rigorous definitions of all DFD data elements and data structures. In our data flow diagrams, we give names to data flows, processes and stores. Although the names are descriptive of the data, they do not give details. So, following the DFD, our interest is to build some structured place to keep details of the contents of the data flows, process and data store.

Name	**Where used**	**Additional Description**
User	In DFD’s	This is an Object on client side.
Login	In DFD’s	This is a Process wherein the user may enter his/her user id and password to get entry to the system.
DashBoard	In DFD’s	This is an Object of the main user interface which controls rest of components of WeaveCloud.
Validate User	In DFD’s	This is a Process wherein the user login details are checked in the database of server for login or not.
Server	In DFD’s	This is the main server side Object
Connect to Server	In DFD’s	It is the Process wherein the DashBoard attempts to connect to the server
User Details	In	This is a Data Store

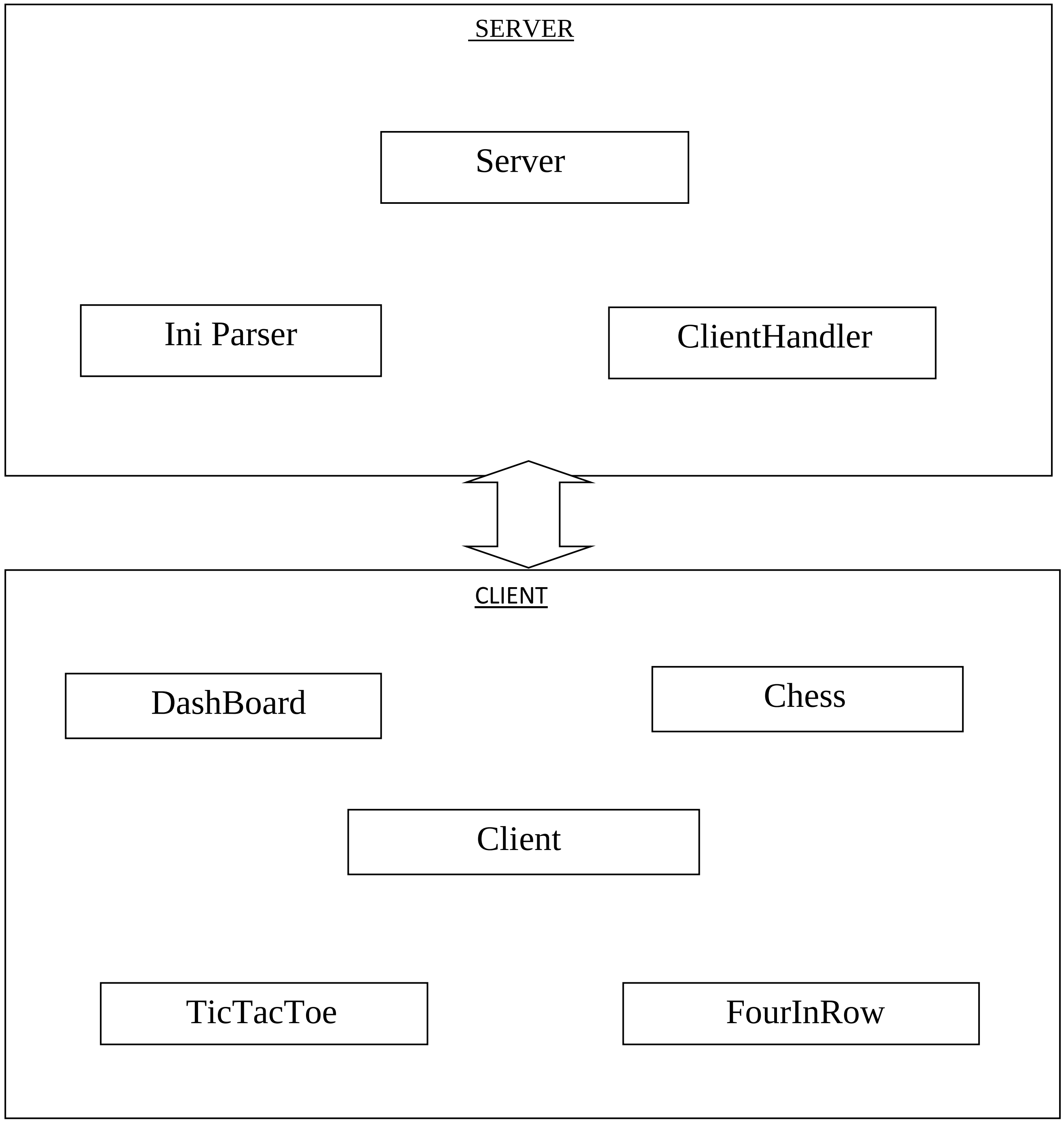
	DFD’s	that keeps a list of users and the details.
Register	In DFD’s	This is a Process in which a new user is registered to WeaveCloud by entering his registration details.
Validate Username and Password	In DFD’s	This is a Process wherein the details entered by user during login i.e. user id and password are checked on server database.
Insert into Database	In DFD’s	In this process, the user details are written into the database on the server.

USE CASE DIAGRAM

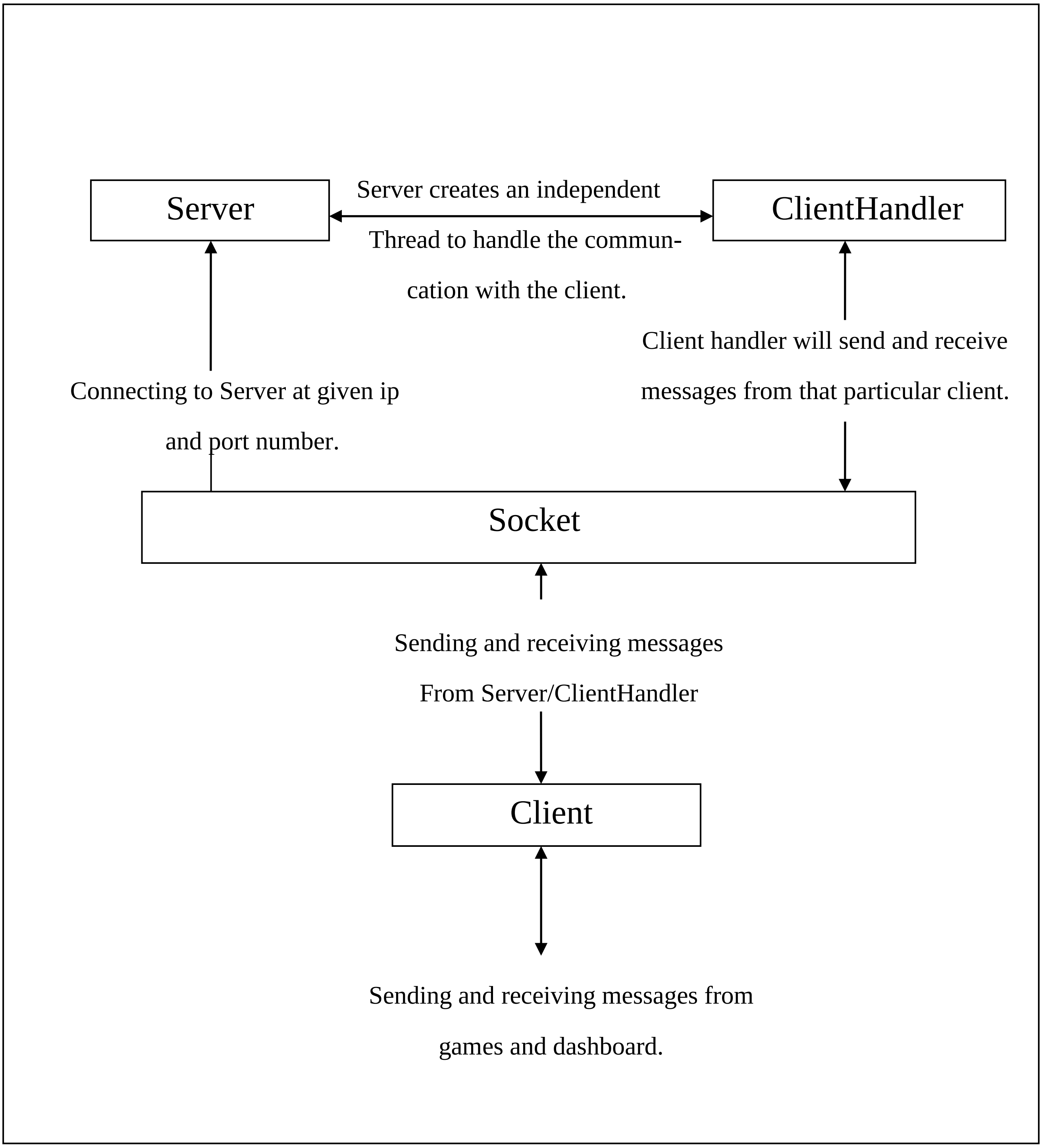


Response

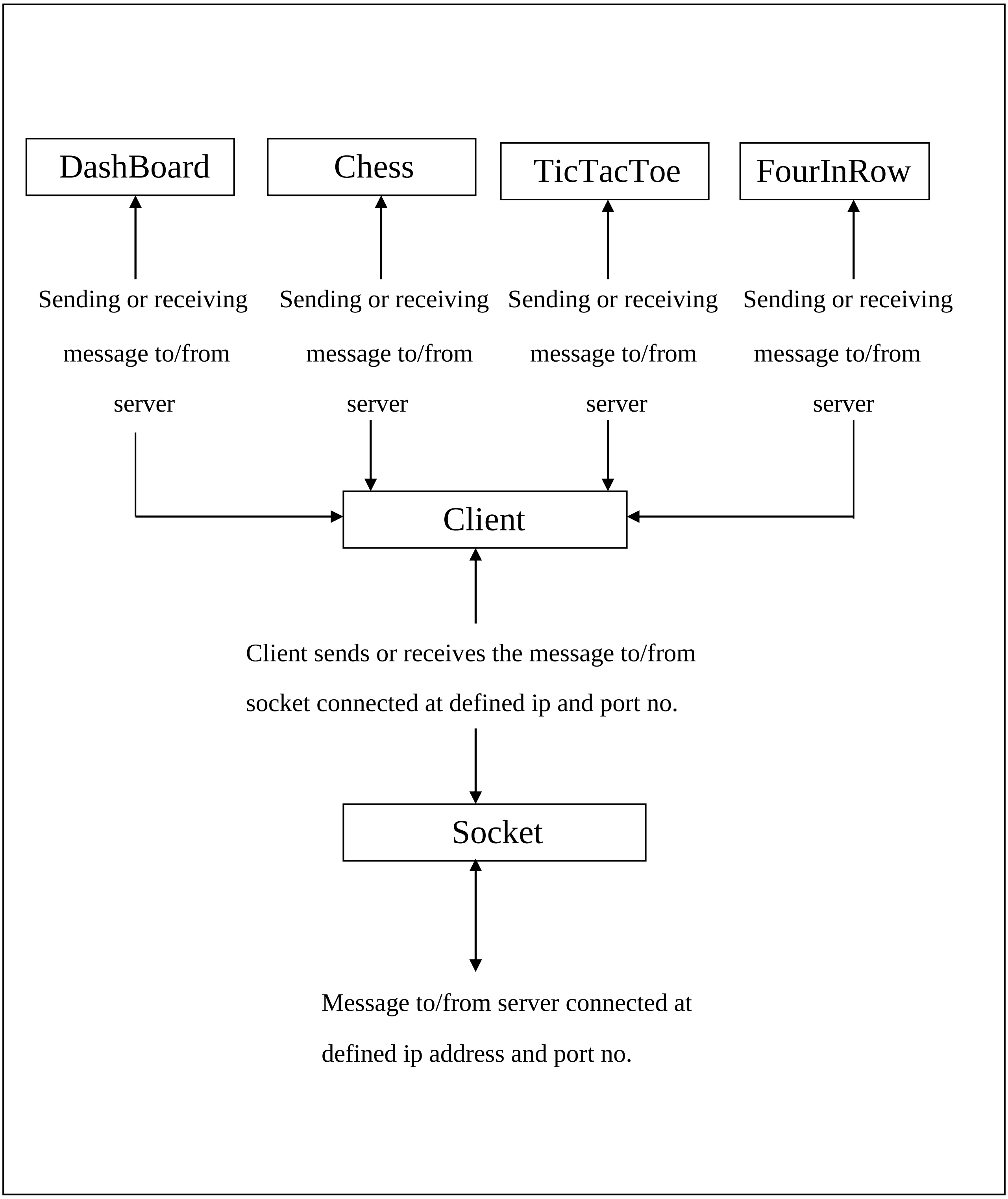
Architecture Diagrams of WeaveCloud



(Fig. shows the basic components both at client side and server side)


Server
Server creates an independent Thread to handle the commun-cation with the client.
ClientHandler
Connecting to Server at given ip and port number.
Client handler will send and receive messages from that particular client.
Socket
Sending and receiving messages From Server/ClientHandler
Client
Sending and receiving messages from games and dashboard.

(Fig. shows the architecture of SERVER)


DashBoard
Chess
TicTacToe
FourInRow
Sending or receiving message to/from server
Sending or receiving message to/from server
Sending or receiving message to/from server
Sending or receiving message to/from server
Client
Client sends or receives the message to/from socket connected at defined ip and port no.
Socket
Message to/from server connected at defined ip address and port no.

(Fig. shows the architecture of CLIENT)