SECURITY-AWARE DYNAMIC SCHEDULING FOR REAL-TIME OPTIMIZATION IN AWS

A PROJECT REPORT

Submitted by

HARINIPRIYA.S   **[REGISTER NO: 211419104091]**

DHARRMIYA THULASI.B   **[REGISTER NO: 211419104062]**

MADHUMITHA.P   **[REGISTER NO: 211419104155]**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2023

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"Security Aware Dynamic Scheduling for Real-Time Optimization in AWS"** is the bonafide work of **"Harinipriya.S (211419104091), Dharrmiya thulasi.B (211419104062), Madhumitha.P (211419104155)"** who carried out the project work under my supervision.

SIGNATURE   **SIGNATURE**

Dr.L.JABASHEELA,M.E.,Ph.D.,   **Ms.V.SATHIYA,M.E**
HEAD OF THE DEPARTMENT   **SUPERVISOR**

DEPARTMENT OF CSE,   DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,   PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,   NASARATHPETTAI,
POONAMALLEE,   POONAMALLEE,
CHENNAI-600 123.   CHENNAI-600 123.

Certified that the above candidate(s) was examined in the End Semester Project Viva-Voce Examination held on..........................

INTERNAL EXAMINER   **EXTERNAL EXAMINER**

DECLARATION

We **Harinipriya.S (211419104091), Dharrmiya thulasi.B (211419104062), Madhumitha.P (211419104155)** hereby declare that this project report titled "**Security Aware Dynamic Scheduling for Real-Time Optimization in AWS**", under the guidance of **Ms.V.SATHIYA.M.E,** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

HARINIPRIYA.S

DHARRMIYA THULASI.B

MADHUMITHA.P

ACKNOWLEDGEMENT

ABSTRACT

The distributed computing is a business and monetary model permitting the clients to use very good quality figuring and stockpiling basically with negligible foundation on their end. The development of safety mindful modern control has prepared of safety mindful booking in cloud-based modern applications. As a matter of fact, most cloud-based modern applications are time touchy, which need constant handling. Edge distributed computing worldview broadens the processing skill of customary cloud model with low-inertness neighborhood assets. In this way, heterogeneous mists that comprise of both unified assets and edge assets might be a promising asset model to give both versatile and low-dormancy assets for cloud-based modern applications. This task executes constant asset access control in modern control frameworks. Three-level security model is intended for cloud assets access in modern control frameworks.

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION:

Cloud-based administrations are applications or programming that are accessible from a distance and facilitated on the seller's server for the benefit of the client. One simple method for seeing this is to in any case consider the product being on-premises, yet it happens to likewise be accessible from a distance. It likewise fills in as a stage that supports early arrangement and planning applications which can utilize the gathered information to foster creation schedules.in this ventures we have proposed to plan for Modern web application.

1.2 SCOPE OF THE PROJECT:

There are many different business models to choose from when it comes to building a Web Application. Few of these business models are mentioned A great project scope example is an effective tool typically used in project management. It is used to explain the most important deliverables of a project. These include the major milestones, top level requirements, assumptions as well as limitations. In this type of business model, the data from all the carriers is accumulated at one point. In this business model, the company requires only a single web application for all the packages regardless of the carrier. Packages can be added seamlessly by using its ID. Moreover, the web application keeps the customer updated regarding the delivery status.

1.3 LITERATURE SURVEY:

TITLE: An Integrated Cloud CAE Simulation System for Industrial Service Applications

AUTHOR NAME: JIACHENG XIE 1,2, XUEWEN WANG1, ZHAOJIAN YANG1, AND SHANGQING HAO2

PAPER YEAR: February 27, 2019.

PAPER EXPLANATION:

The cloud-based computer-aided engineering (CAE) technology expands the application scope of CAE, solves the problem of uneven distribution, and improves the efficiency of the simulation resources. Consequently, various industries are turning to cloud CAE technology to design their products. However, to utilize this technology fully, we must overcome problems, such as impracticality, inability to handle complex analysis objects, poor data transmission, and compatibility, and poor industrial applicability. Based on the ASP.NET, VB.NET, and ANSYS Parametric Design Language, this paper establishes an integrated cloud CAE simulation system for industrial service applications. The system: 1) connects the dispatching manager through the network and 2) clusters the servers according to their load and usage before dispatch, thereby enabling users to access the ANSYS software remotely. The model can be built either by inputting the dimensions of the parametric model through the page or by uploading a three-dimensional computer aided design (CAD) model that meets the format requirements. The material parameters and the boundary conditions also input through the page. Thus, clients can input their own parameters and analyze the server CAD. Finally, the CAE analysis results can be invoked by a page operation, saving meaningful results on the server to be accessed directly by users. The proposed simulation system serves the needs of users both quickly and efficiently. In the large-scale industrial application trials, enterprises responded favorably to the system. In particular, it properly solves the CAE analysis problems of small- and medium-sized machinery and equipment enterprises.

TITLE: Cloud-Based Cyber-Physical Robotic Mobile Fulfillment Systems: A Case Study of Collision Avoidance

AUTHOR NAME: K. L. KEUNG, C. K. M. LEE

PAPER YEAR: May 22, 2020.

PAPER EXPLANATION: The rapid development and implementation of the Internet of Things (IoT) and Cyber-Physical Systems (CPS) in the engineering and manufacturing field have embraced a virtual identity to ensure nearly real-time adjustment. Warehouses are challenged to reassess its order fulfillment operations while simultaneously being provided with the opportunity to develop its own cloud-based CPS with the aid of IoT devices. Robotic Mobile Fulfillment System (RMFS) is a system controlling mobile robots, mobile storage rack, put away and picking workstations, charging stations, and wireless communication infrastructure in the context of robotic-assisted warehouse. This paper addresses the value creation utilizing cloud-based CPS in RMFS. By providing an analysis of cloud services and IoT enhancement, theoretical concepts from the literatures are consolidated to solve the research que-stions on how RMFS offering better order fulfillment can gain benefits in terms of operational efficiency and system reliability. The paper also proposes a cloud based CPS architecture, providing a comprehensive understanding on conflict avoidance strategy in the multi-layers multi-deeps warehouse layout. This research presents six conflict classifications in RMFS and provides a case study in the real-life context. Dock grid conflict is a new type of conflict appearing in multideeps RMFS. A scenario analysis with real customer orders is applied to present the collision detection and solution.

TITLE: Delay Mitigation in Offloaded Cloud Controllers in Industrial IoT

AUTHOR NAME: SAAD MUBEEN, ALMA DIDIC, KRISTIAN SANDSTRÖM

PAPER YEAR: April 24, 2017.

PAPER EXPLANATION: This paper investigates the interplay of cloud computing, fog computing, and Internet of Things (IoT) in control applications targeting the automation industry. In this context, a prototype is developed to explore the use of IoT devices that communicate with

a cloud-based controller, i.e., the controller is offloaded to cloud or fog. Several experiments are performed to investigate the consequences of having a cloud server between the end device and the controller. The experiments are performed while considering arbitrary jitter and delays, i.e., they can be smaller than, equal to, or greater than the sampling period. This paper also applies mitigation mechanisms to deal with the delays and jitter that are caused by the networks when the controller is offloaded to the fog or cloud.

TITLE: Architecture for a Multimodal and Domain-Independent Clinical Decision Support System Software Development Kit

AUTHOR NAME: Naiara Muro, Jordi Torres

PAPER YEAR: 23-27 July 2019

PAPER EXPLANATION: Digitalization of the decision-making process in healthcare has been promoted to improve clinical performance and patient outcomes. The implementation of Clinical Practice Guidelines (CPGs) using Clinical Decision Support Systems (CDSSs) is widely developed in order to achieve this purpose within clinical information systems. Nevertheless, due to several factors such as (i) incompleteness of CPG clinical knowledge, (ii) out-of-date contents, or (iii) knowledge gaps for specific clinical situations, guideline-based CDSSs may not completely satisfy clinical needs. The proposed architecture aims to cope with guideline knowledge gaps and pitfalls by harmonizing different modalities of decision support (i.e. guideline-based CDSSs, experience-based CDSSs, and data mining-based CDSSs) and information sources (i.e. CPGs and patient data) to provide the most complete, personalized, and up-to-date propositions to manage patients. We have developed a decisional event structure to retrieve all the information related to the decision-making process. This structure allows the tracking, computation, and evaluation of all the decisions made over time based on patient clinical outcomes. Finally, different user-friendly and easy-to-use authoring tools have been implemented within the proposed architecture to integrate the role of clinicians in the whole process of knowledge generation and validation. A use case based on Breast Cancer management is presented to illustrate the performance of the implemented architecture

TITLE: Time and Attribute Based Dual Access Control and Data Integrity Verifiable Scheme in Cloud Computing Applications

AUTHOR NAME: DUO ZHANG, JIFANG WANG

PAPER YEAR: October 3, 2019.

PAPER EXPLANATION: In the era of big data, cloud-based industrial applications can provide available and convenient data access for resource-constrained smart devices. Attribute-based encryption can be used to ensure data security while providing fine-grained data access. However, current attribute-based encryption schemes rarely consider the access control of time, and the integrity verification of data simultaneously. In response to the above two problems, we propose a time and attribute based dual access control and data integrity verifiable scheme in cloud computing applications (DCDV). Firstly, a hierarchical time tree is introduced in the attribute-based encryption technology by using of hierarchical identity-based encryption technology to set an effective access time period and a specified decryptable time period for the user's attributes key and encrypted data separately. The decryption operation can only be performed if the attribute set of user satisfies the data owner's access policy and the effective access time period of the user's attributes key completely covered the decryption time period set by the data owner. In this way, the data is dual controlled with time and attributes to solve the problem of privacy data leakage caused by private key leakage. Secondly, by using of the inverted index and Merkle hash tree, the data verification tree is designed. The data user can verify the integrity of the ciphertext data returned by the cloud server without decryption, which solves the problem that the cloud server may delete or modify the data. Finally, the security proof and efficiency analysis show that our scheme is secure and practical.

1.4 EXISTING AND PROPOSED SYSTEM:

EXISTING SYSTEM	PROPOSED SYSTEM
When resources are allocated to the concerned users separately, require more resources. Though a particular number of resources are available, want to schedule the accessibility. Sometimes, resources are allocated there is a change to access by unknown users.	Whenever need those, daily request wants to provide to an authorized person. After finishing their services, the authority will remove the access control to the members. They provide access control again to them when receiving the request from the users or concerned persons.
Technique: PSO Algorithm.	**Technique:** RSA algorithm. SQL techniques.
Disadvantage: It is easy to fall into local optimum in high-dimensional space and has a low convergence rate in the iterative process.	**Advantage:** It provides default robust security and resilient procedures.

CHAPTER 2
PROJECT DESCRIPTION

CHAPTER 2

PROJECT DESCRIPTION

2.1 GENERAL

Thus, there search database used by the doctor is anonymous. Suppose that certain data concerning visitors are related to the use of a drug over a period of four years and certain side effects have been observed and recorded by the doctors in the doctor's database. It is clear that these data (even if anonymized) need to be kept confidential and accessible only to the few specialists of the institution working on this project, until further evidence is found about the drug. If these anonymous data were to be disclosed, privacy of the visitors would not be at risk; however, the company manufacturing the drug may be adversely affected.

2.2 PROBLEM DEFINITION:

It is very important to maintain efficient software to handle information of a hospital. This application provides *a way* to record this information and to access these in a simple way.

2.3 METHODOLOGIES

Methodologies is the process of analyzing the principles or procedure of a Progressive Anonymous Database management system.

2.4 MODULES

- ➢ LOGIN
- ➢ ADD DETAILS
- ➢ PICKUP
- ➢ DELIVERY
- ➢ REPORT UPLOAD
- ➢ VIEW
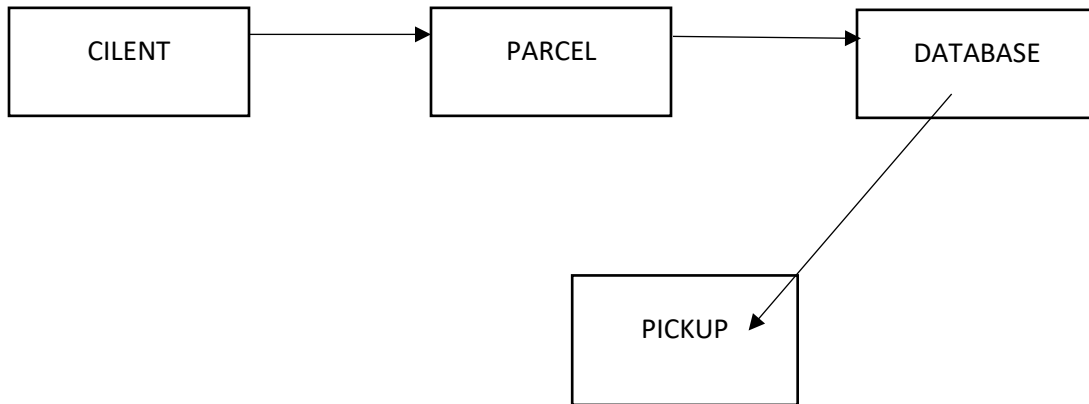
MODULE DISCRIPTION:

LOGIN:

This module gives the way to enter into main page after login with valid input such as username or email id and password.

ADD DEATAILS:

This module gives the way to client enter the details in form. Fill the To address.

PICKUP:

This module picker up collect the client parcels.

DELIVERY:

This module is used to delivery client parcel in send address.

REPORT UPLOAD:

Branch managers upload the daily branch report to head office.

VIEW:

This module to help us head officer view the daily branch report. Head office view the branch manager

LOGIN

```
┌──────────┐        ┌──────────┐        ┌──────────────┐
│  CLIENT  │───────▶│  LOGIN   │───────▶│ Visitor Main │
└──────────┘        └──────────┘        └──────────────┘
```

ADD DETAILS:

```
┌──────────┐        ┌──────────┐        ┌──────────────┐
│   FORM   │───────▶│ DETAILS  │───────▶│   DATABASE   │
└──────────┘        └──────────┘        └──────────────┘
```

PICKUP:

```
┌──────────┐        ┌──────────┐        ┌──────────┐
│  CILENT  │ ─────► │  PARCEL  │ ─────► │ DATABASE │
└──────────┘        └──────────┘        └──────────┘
                                              │
                                              ▼
                                   ┌──────────┐
                                   │  PICKUP  │
                                   └──────────┘
```

DELIVERY:

```
┌────────────┐        ┌────────────┐        ┌──────────────┐
│ PICKER HUB │ ─────► │  DATABASE  │ ─────► │ DELIVERY HUB │
└────────────┘        └────────────┘        └──────────────┘
```

REPORT UPLOAD

```
┌──────────┐        ┌──────────┐        ┌──────────┐
│  BRANCH  │ ─────► │  UPLOAD  │ ─────► │ DATABASE │
│ MANAGER  │        └──────────┘        └──────────┘
└──────────┘
```

VIEW

```
┌──────────┐        ┌──────────┐        ┌──────────┐
│   HEAD   │ ─────► │ DATABASE │ ─────► │   VIEW   │
└──────────┘        └──────────┘        └──────────┘
```

15

2.5 ALGORITHM USED:

1.RSA Algorithm:

The RSA algorithm is an asymmetric cryptography algorithm; this means that it uses a public key and a private key (i.e two different, mathematically linked keys). As their names suggest, a public key is shared publicly, while a private key is secret and must not be shared with anyone.

The RSA algorithm is named after those who invented it in 1978: Ron Rivest, Adi Shamir, and Leonard Adleman.

How does RSA work?

- RSA (Rivest-Shamir-Adleman) is an asymmetric cryptographic algorithm
used to encrypt and decrypt messages by modern computers. Asymmetric states that there are two different keys used in the encryption and decryption process, which also is called public-key cryptography.
- This is simply because one of the two keys can be given to anyone without
exploiting the security of the algorithm.
- The RSA algorithm involves both private and public keys. The public key
can be known and published to anyone, as it is used to encrypt the messages from plaintext to ciphertext. The messages that are encrypted with this specific public key can however only be decrypted with the corresponding private key. The key
generation process of the RSA algorithm is what makes it so secure and reliable today, as it contains a high level of complexity compared to other cryptographic algorithms.

KEY GENERATION

Unlike symmetric algorithms, such as for example AES, public key algorithms require the computation of the pair (K_{public}, Kprivate). What makes RSA so special compared to other encryption algorithms is that these keys must be computed using mathematics and are not random numbers that are generated.

The key generation part of the RSA algorithm is quite central and important, and this is something that's missing in most symmetric key algorithms, where the key generation part is not really complicated in terms of mathematical computations.

The key generation process of the RSA algorithm consists of five steps:

1. Choose two large prime numbers (p and q)

2. Compute $n = p \cdot q$

3. Calculate $\varphi(n) = (p - 1) \cdot (q - 1)$

4. Choose an integer e such that $1 < e < \varphi(n)$, and:

(a) Ensure that $\gcd(e, \varphi(n)) = 1$

(b) Ensure that e and $\varphi(n)$ are coprime

5. Compute an integer d, such that $d = e^{-1} \mod \varphi(n)$

After completing these five steps, we have generated two asymmetric keys that can be used for encryption and decryption further in the process: the public key consists of n and e, while the private key consists of d.

$K_{public} = (n, e)$

$K_{private} = (d)$

It is a common practice to use large numbers in the generation process for p and q, often at least 512 bits, which makes the n 1024 bits ($n = p \cdot q$). This increases the level of complexity and makes the cryptanalysis process considerable harder in terms of brute force attack.

2. PSO Algorithm:

Particle swarm optimization (PSO) is a stochastic population based metaheuristic algorithm, inspired from the behavior of fish schooling and flocking of birds. It was first developed by Kennedy and Eberhert in 1995, as a method to simulate social behavior based on bird flocking and fish schooling.

How does work?

Since the introduction of the PSO several variations have been presented. Here

we will only explain in detail how the original one works and then summarize two variants. For more in-depth information on the variants please see references [1, 6, 3]. Remember that the main concept is that we have particles of a swarm moving in a problem space and evaluating their positions through a fitness function. Once a problem space is defined a set of particles is spawned in it and their positions and velocities are updated iteratively according to the specific PSO algorithm. Even though PSO has been proven to be an efficient algorithm with good results it is not by design one that guarantees that the best solution is found, since it relies on visiting and evaluating problem space positions. Even though many variations exist usually all have a fitness function. The specification of this function depends on the problem being optimized (especially in its dimensions) and as such we will simply refer to it as

$f(x_i)$ being the

short for $f(x_{i,0}), ..., f(x_{i,d})$. This function represents how good the i particle's position in the multidimensional space is relatively to the desired goal. With this it weights and binds the D dimensions to be optimized, given a problem modeled as an optimization one of d dimensions. By being a multi-dimensional algorithm, the positions and velocities of the particles we manipulated will then have d components, so we have positions as x_i

Global Best (original version):

In this algorithm we have a completely connected swarm, meaning that all the particles share information, any particle knows what the best position is ever visited by any particle in the swarm.

CHAPTER 3

REQUIREMENT ENGINEERING

CHAPTER 3

REQUIREMENTS ENGINEERING

3.1 GENERAL

These are the requirements for doing the project. Without using these tools and software's we cannot do the project. So, we have two requirements to do the project. They are,

1. Hardware Requirements.
2. Software Requirements.

3.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

PROCESSOR	:	PENTIUM IV 2.6 GHz, Intel Core 2 Duo.
RAM	:	4GB DD RAM
MONITOR	:	15" COLOR
HARD DISK	:	40 GB

3.3 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks, and tracking the team's and tracking the team's progress throughout the development activity.

|		
Front End	:	J2EE (JSP, SERVLETS) JAVASCRIPT
Back End	:	MY SQL 5.5
Operating System	:	Windows 07
IDE	:	Eclipse

20

CHAPTER 4

DESIGN

ENGINEERING

CHAPTER 4

DESIGN ENGINEERING

4.1 GENERAL

Design Engineering deals with the various UML [Unified Modeling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

4.1.1 USE CASE:
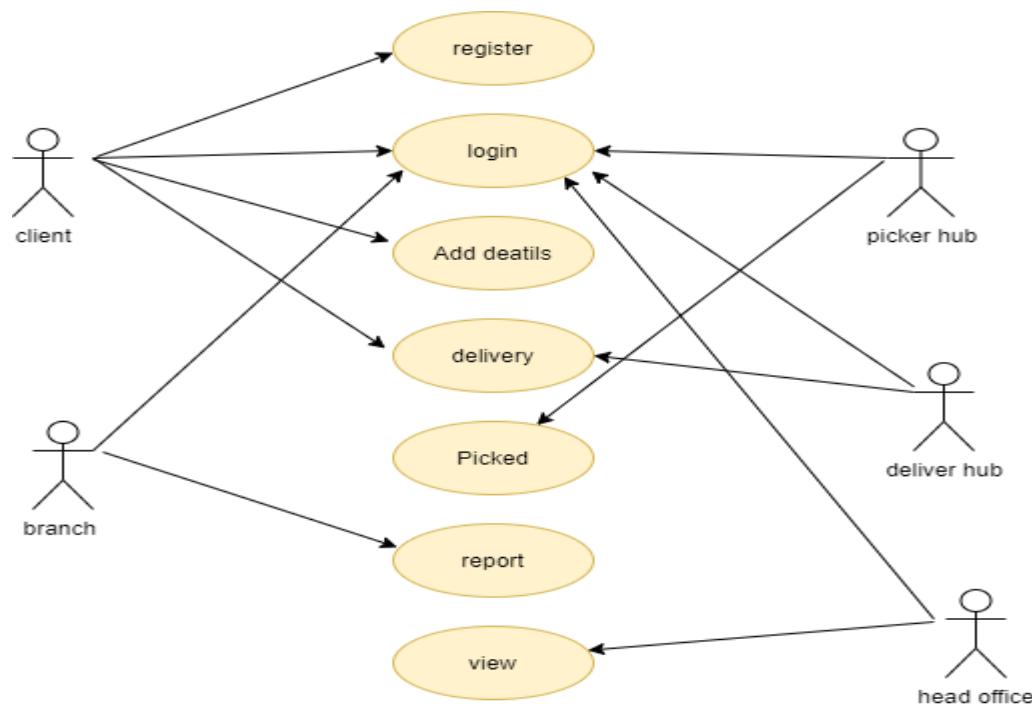
EXPLANATION:

The use case diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. For this in our component diagram first

propose a data In this proposed method we are using Hash-Solomon Code Algorithm to encrypt the data.

4.1.2 STATE DIAGRAM:
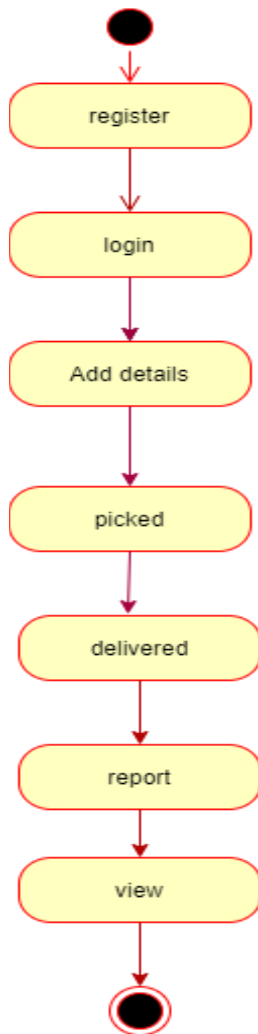
EXPLANATION:

State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics. In our state diagram first propose a . For this in our component diagram first propose a data In this proposed method we are using Hash-Solomon Code Algorithm to encrypt the data.
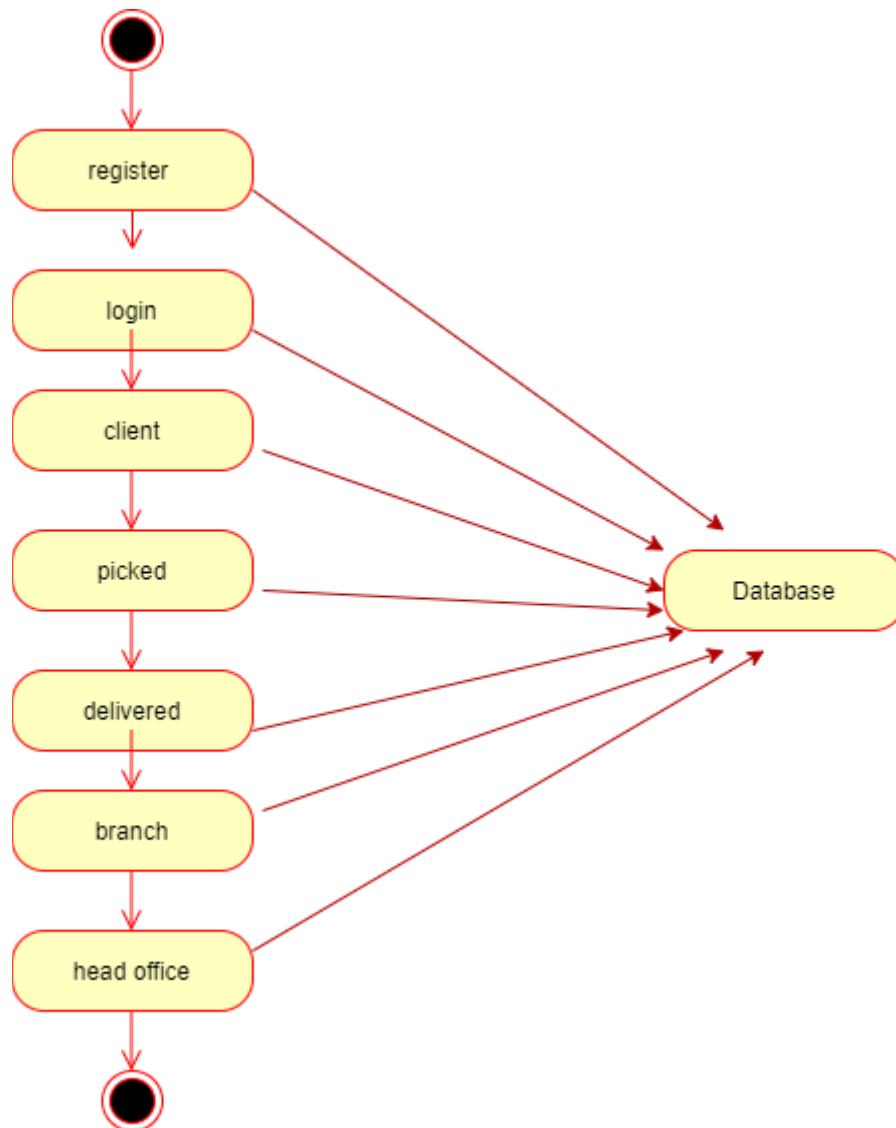
4.1.3 ACTIVITY DIAGRAM

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.

4.1.4 CLASS DIAGRAM:

Client
register
login
checkout
username();;
password();
add();

Picker
login
view
username();
password();
picked();

Branch
login
view
uploa
username();
password();
upload();

Delivery hub
login
view
delivery
username();
password();
delivery

Headoffice
login
view
username();
password();
view();

EXPLANATION:

Class diagram 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. The classes in a class diagram represent both the main objects and or interactions in the application and the objects.

4.1.5 SEQUENCE DIAGRAM:
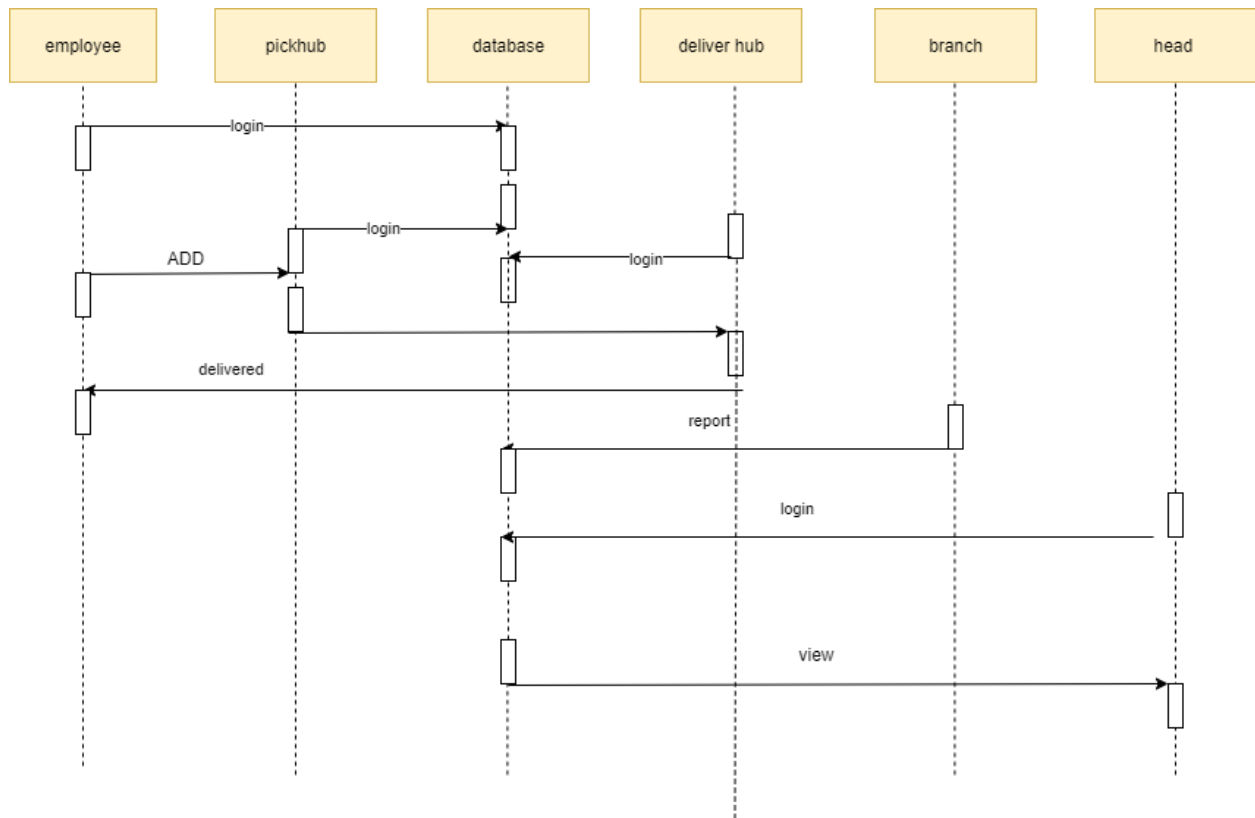
EXPLANATION:

In our sequence diagram specifying processes operate with one another and in order. In our sequence diagram first propose a for this in our component diagram first propose a data In this proposed method we are using Hash-Solomon Code Algorithm to encrypt the data.
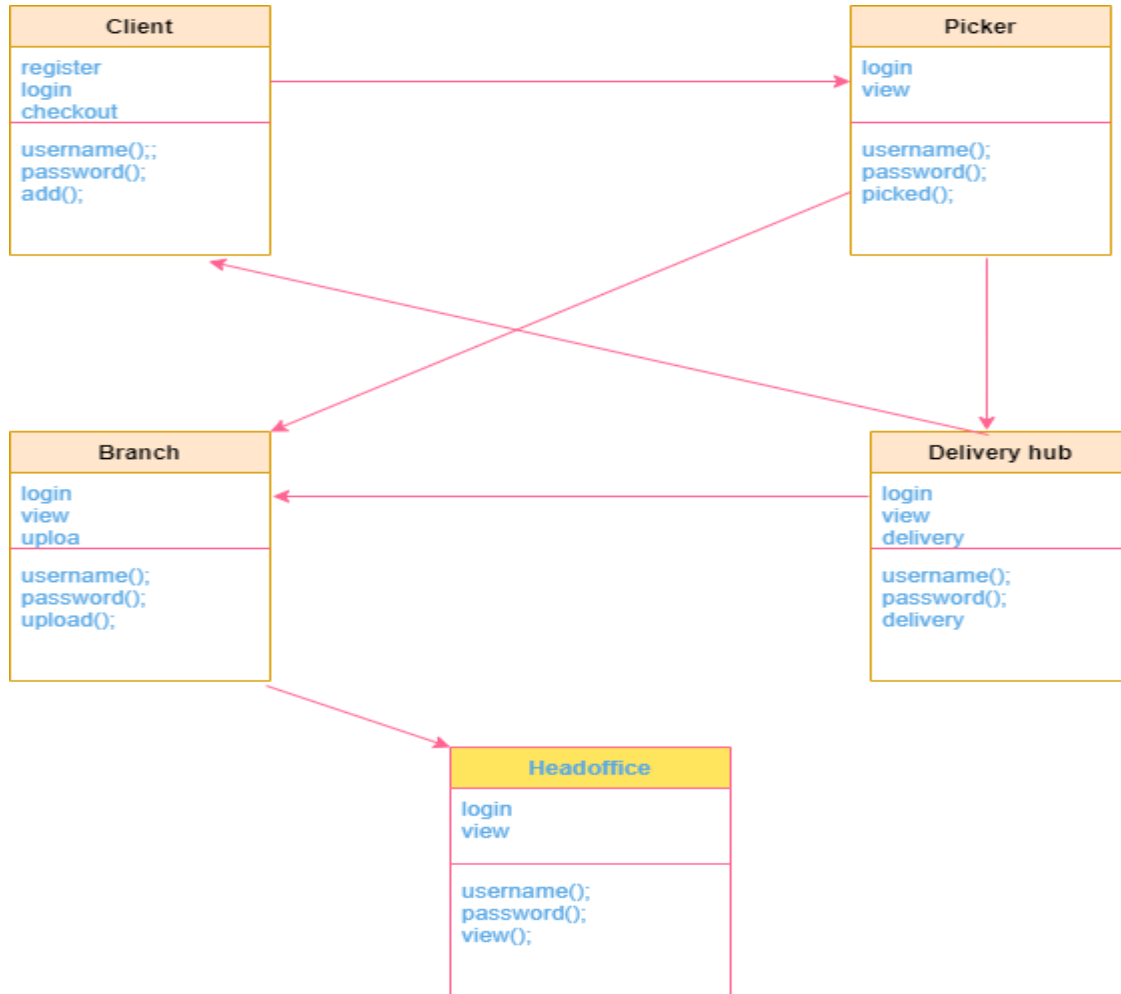
4.1.6 COLLABORATION DIAGRAM

A collaboration diagram shows 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).
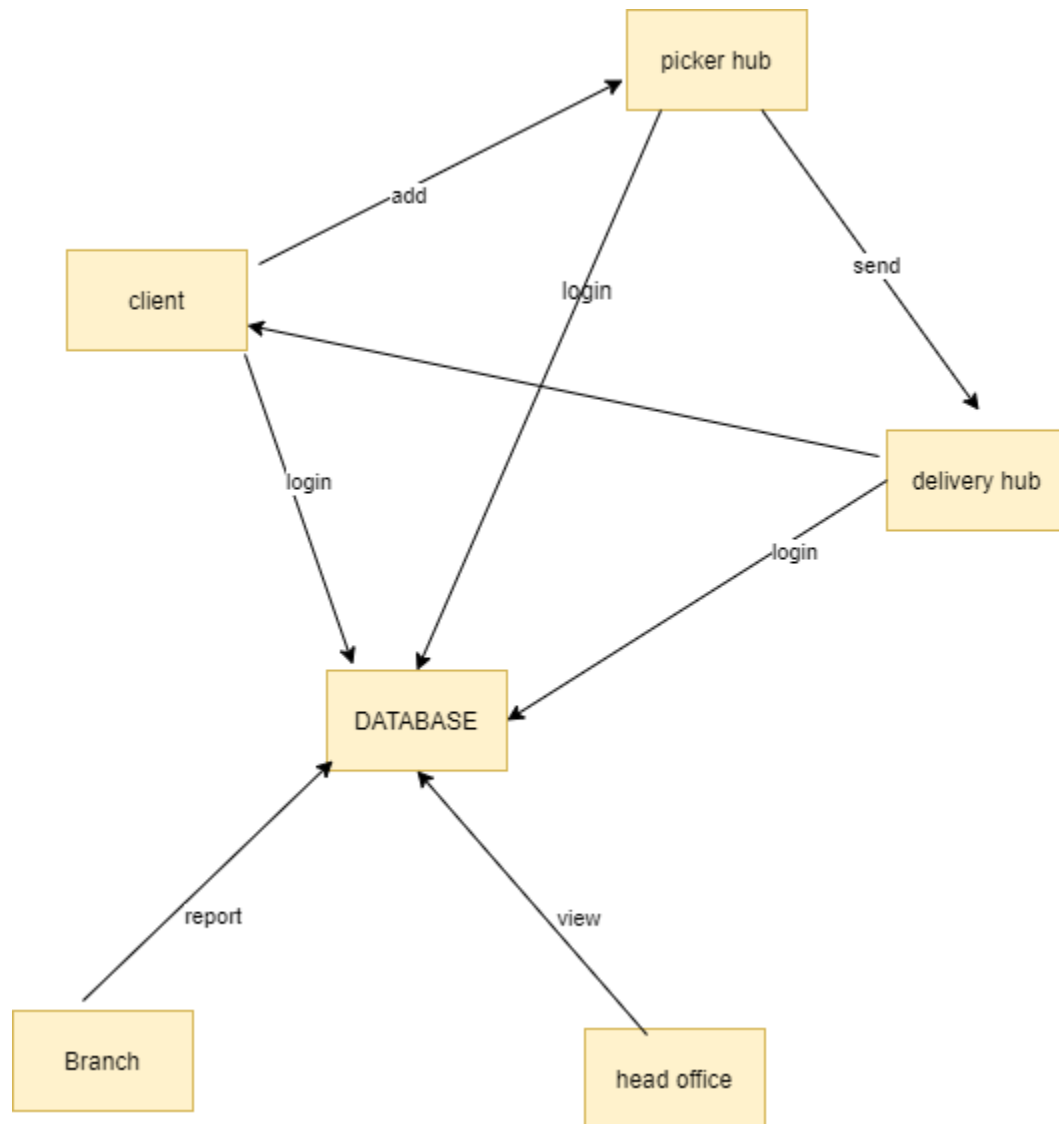
4.1.7 COMPONENT DIAGRAM:
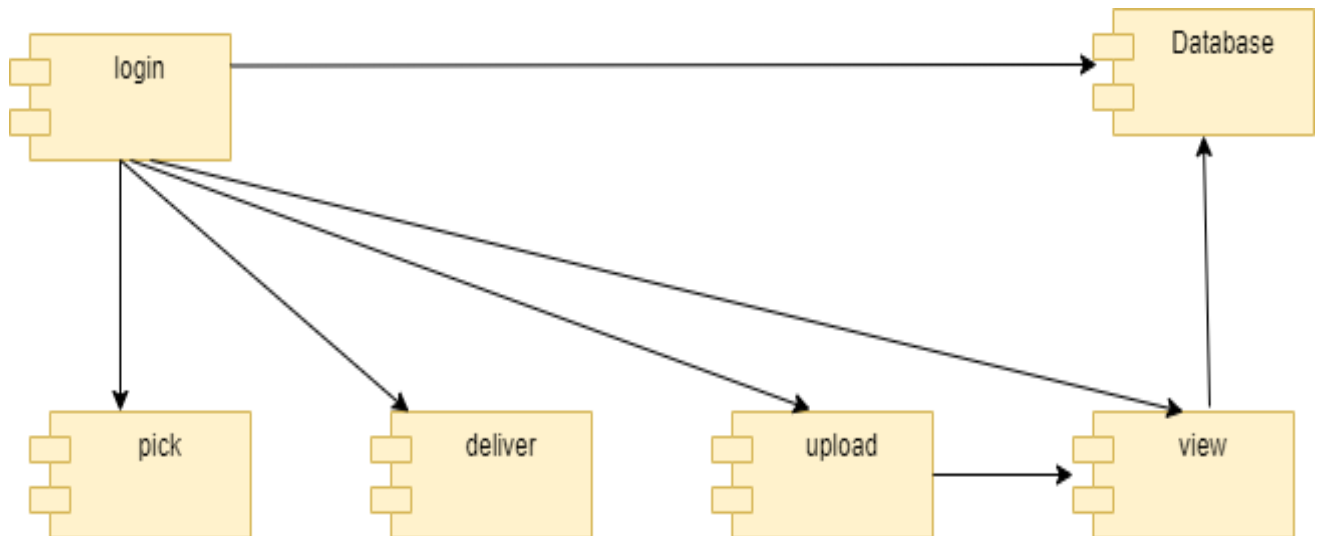
A component diagram displays the structural relationship of components of a software system. These are mostly used when working with complex systems that have many components such as sensor nodes, cluster head and base station. It does not describe the functionality of the system, but it describes the components used to make those functionalities. The interfaces are linked using connectors.

4.1.8 DATAFLOW 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.

LEVEL 0:

LEVEL 1:
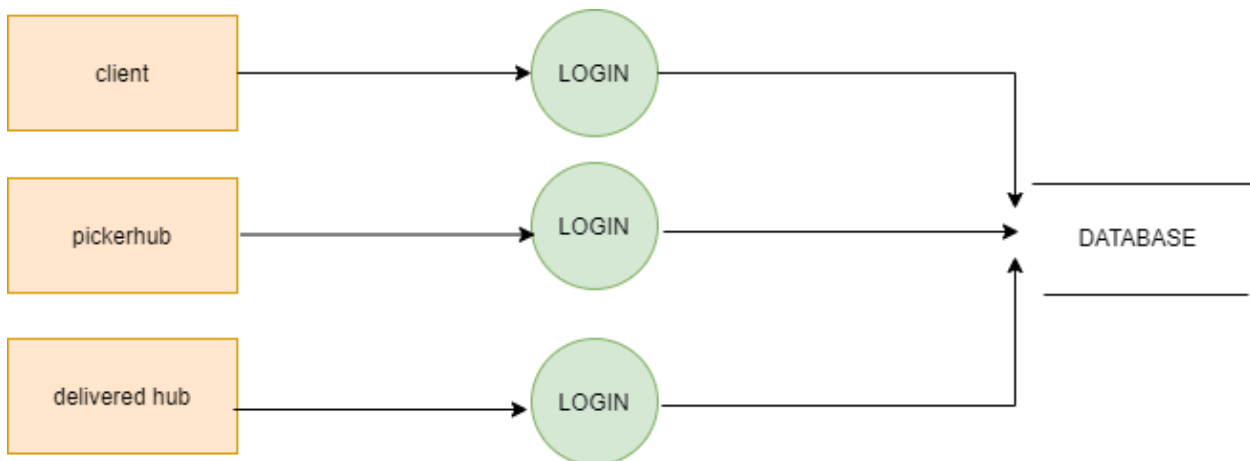
Level 2:
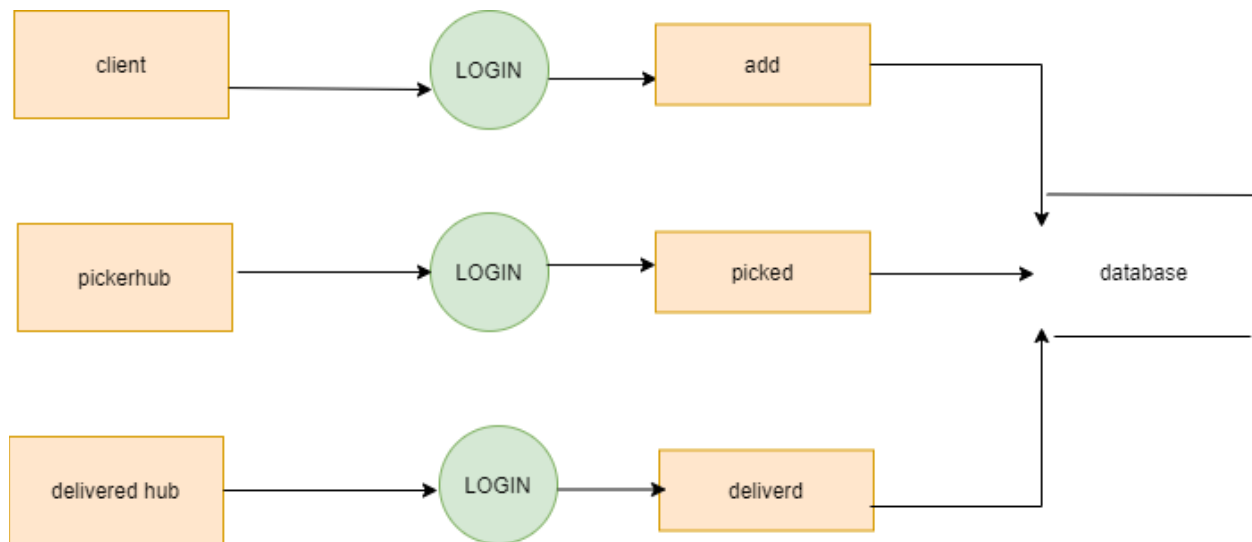
4.1.9 ER-DIAGRAM:
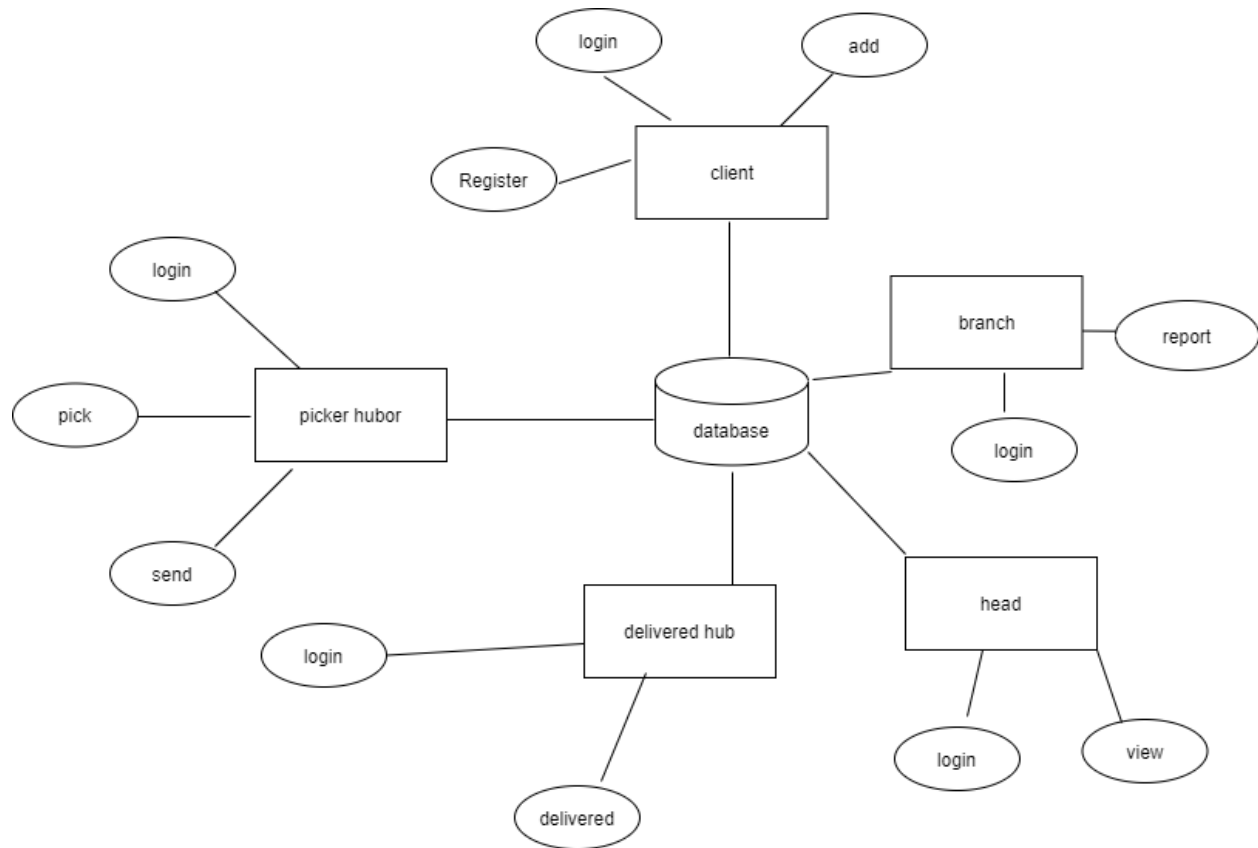
EXPLANATION:

An entity is represented as rectangle in an ER diagram. For example: In the following ER diagram we have two entities Student and College, and these two entities have many to one relationship as many students' studies in a single college. We will read more about relationships later, for now focus on entities.

4.1.10 SYSTEM ARCHITECTURE:
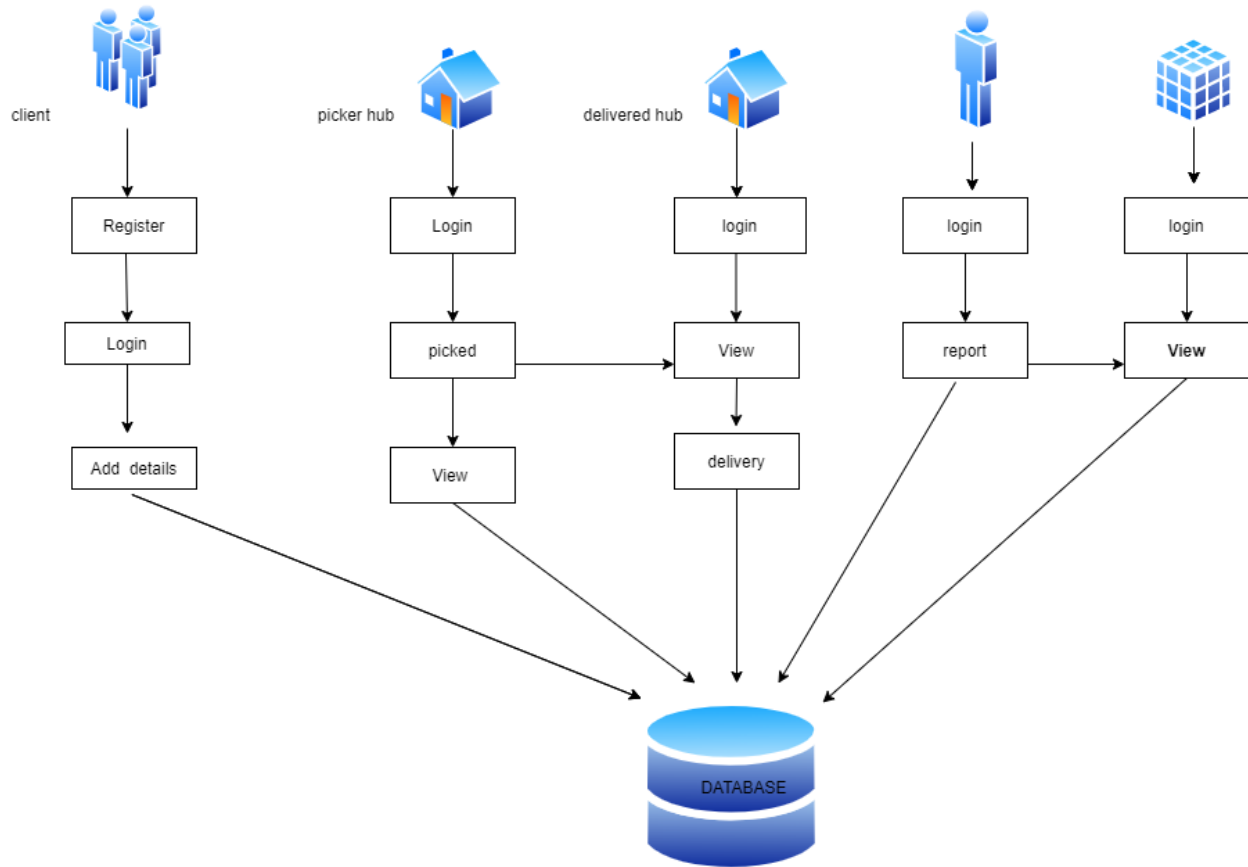
EXPLANATION:

The systems architect establishes the basic structure of the system, we propose a Hash code Solomon algorithm and a we can put a small part of data in local machine and fog server in order to protect the privacy. Moreover, based on computational intelligence, this algorithm can compute the distribution proportion stored in cloud, fog, and local machine, respectively. Through the theoretical safety analysis and experimental evaluation, the feasibility of our scheme has been validated, which is really a powerful supplement to existing cloud storage scheme

CHAPTER 5
SOFTWARE SPECIFICATION

CHAPTER 5
SOFTWARE SPECIFICATION

5.1 GENERAL:

This chapter is about the software language and the tools used in the development of the project. The platform used here is JAVA. The Primary languages are JAVA, J2EE and J2ME. In this project J2EE is chosen for implementation.

5.2 FEATURES OF JAVA

5.2.1 THE JAVA FRAMEWORK

Java is a programming language originally developed by James Gosling at Sun Microsystems and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is general-purpose, concurrent, class-based, and object-oriented, and is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere".

Java is considered by many as one of the most influential programming languages of the 20th century and is widely used from application software to web applications The java framework is a new platform independent that simplifies application development internet. Java technology's versatility, efficiency, platform portability, and security make it the ideal technology for network computing. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

5.2.2 OBJECTIVES OF JAVA

To see places of Java in Action in our daily life, explore java.com.

WHY SOFTWARE DEVELOPERS CHOOSE JAVA:

Java has been tested, refined, extended, and proven by a dedicated community. And numbering more than 6.5 million developers, it's the largest and most active on the planet. With its versatility, efficiency, and portability, Java has become invaluable to developers by enabling them to:

- Write software on one platform and run it on virtually any other platform
- Create programs to run within a Web browser and Web services
- Develop server-side applications for online forums, stores, polls, HTML forms processing, and more
- Combine applications or services using the Java language to create highly customized applications or services
- Write powerful and efficient applications for mobile phones, remote processors, low-cost consumer products, and practically any other device with a digital heartbeat

SOME WAYS SOFTWARE DEVELOPERS LEARN JAVA:

Today, many colleges and universities offer courses in programming for the Java platform. In addition, developers can also enhance their Java programming skills by reading Sun's java.sun.com Web site, subscribing to Java technology-focused newsletters, using the Java Tutorial and the New to Java Programming Center, and signing up for Web, virtual, or instructor-led courses.

Object-Oriented:

To be an Object-Oriented language, any language must follow at least the four characteristics.

1. Inheritance: It is the process of creating the new classes and using the behavior of the existing classes by extending them just to reuse the existing code and adding addition a feature as needed.

2. Encapsulation: It is the mechanism of combining the information and providing the abstraction.

3. Polymorphism: As the name suggest one name multiple form, Polymorphism is the way of providing the different functionality by the functions having the same name based on the signatures of the methods.

4. Dynamic binding: Sometimes we don't have the knowledge of objects about their specific types while writing our code. It is the way of providing the maximum functionality to a program about the specific type at runtime.

5.2.3 Java Server Pages - An Overview

Java Server Pages or JSP for short is Sun's solution for developing dynamic web sites. JSP provide excellent server-side scripting support for creating database driven web applications. JSP enable the developers to directly insert java code into jsp file, this makes the development process very simple, and its maintenance also becomes very easy.

JSP pages are efficient, it loads into the web server's memory on receiving the request very first time and the subsequent calls are served within a very short period of time.

In today's environment most web sites server's dynamic pages based on user request. Database is very convenient way to store the data of users and other things. JDBC provide excellent database connectivity in heterogeneous database environment. Using JSP and JDBC its very cc easy to develop database driven web application.

Java is known for its characteristic of "write once, run anywhere." JSP pages are plaf Java Server Pages

Java Server Pages (JSP) technology is the Java platform technology for delivering dynamic content to web clients in a portable, secure and well-defined way. The Java Server Pages specification extends the Java Servlet API to provide web application developers with a robust framework for creating dynamic web content on the server using HTML, and XML templates, and Java code, which is secure, fast, and independent of server platforms.

JSP has been built on top of the Servlet API and utilizes Servlet semantics. JSP has become the preferred request handler and response mechanism. Although JSP technology is going to be a powerful successor to basic Servlets, they have an evolutionary relationship and can be used in a cooperative and complementary manner.

Servlets are powerful and sometimes they are a bit cumbersome when it comes to generating complex HTML. Most servlets contain a little code that handles application logic and a lot more code that handles output formatting. This can make it difficult to separate and reuse portions of the code when a different output format is needed. For these reasons, web application developers turn towards JSP as their preferred servlet environment.

5.2.4 Evolution of Web Applications:

Over the last few years, web server applications have evolved from static to dynamic applications. This evolution became necessary due to some deficiencies in earlier web site design. For example, to put more of business processes on the web, whether in business-to-consumer (B2C) or business-to-business (B2B) markets, conventional web site design technologies are not enough. The main issues, every developer faces when developing web applications, are:

1. Scalability - a successful site will have more users and as the number of users is increasing fatly, the web applications have to scale correspondingly.

2. Integration of data and business logic - the web is just another way to conduct business, and so it should be able to use the same middle-tier and data-access code.

3. Manageability - web sites just keep getting bigger and we need some viable mechanism to manage the ever-increasing content and its interaction with business systems.

4. Personalization - adding a personal touch to the web page becomes an essential factor to keep our customer coming back again. Knowing their preferences, allowing them to configure the information they view, remembering their past transactions or frequent search keywords are all important in providing feedback and interaction from what is otherwise a fairly one-sided conversation.

Apart from these general needs for a business-oriented web site, the necessity for new technologies to create robust, dynamic, and compact server-side web applications has been realized. The main characteristics of today's dynamic web server applications are as follows:

1. Serve HTML and XML, and stream data to the web client

2. Separate presentation, logic, and data

3. Interface to databases, other Java applications, CORBA, directory, and mail services

4. Make use of application server middleware to provide transactional support.

5. Track client sessions

5.2.5 Benefits of JSP

One of the main reasons why the Java Server Pages technology has evolved into what it is today, and it is still evolving is the overwhelming technical need to simplify application design by separating dynamic content from static template display data. Another benefit of utilizing JSP is that it allows to more cleanly separate the roles of web application/HTML designer from a software developer. The JSP technology is blessed with several exciting benefits, which are chronicled as follows:

1. The JSP technology is platform independent, in its dynamic web pages, its web servers, and its underlying server components. That is, JSP pages perform perfectly without any hassle on any platform, run on any web server, and web-enabled application server. The JSP pages can be accessed from any web server.

2. The JSP technology emphasizes the use of reusable components. These components can be combined or manipulated towards developing more purposeful components and page design. This definitely reduces development time apart from the At development time, JSPs are very different from Servlets, however, they are precompiled into Servlets at run time and executed by a JSP engine which is installed on a Web-enabled application server such as BEA WebLogic and IBM WebSphere.

5.3 Servlets:

Earlier in client- server computing, each application had its own client program and it worked as a user interface and need to be installed on each user's personal computer. Most web applications use HTML/XHTML that are mostly supported by all the browsers and web pages are displayed to the client as static documents.

A web page can merely display static content and it also lets the user navigate through the content, but a web application provides a more interactive experience.

Any computer running Servlets or JSP needs to have a container. A container is nothing but a piece of software responsible for loading, executing, and unloading the Servlets and JSP. While servlets can be used to extend the functionality of any Java- enabled server.

They are mostly used to extend web servers and are efficient replacement for CGI scripts. CGI was one of the earliest and most prominent server-side dynamic content solutions, so before going forward it is very important to know the difference between CGI and the Servlets.

5.4 Java Servlets

Java Servlet is a generic server extension that means a java class can be loaded dynamically to expand the functionality of a server. Servlets are used with web servers and run inside a Java Virtual Machine (JVM) on the server, so these are safe and portable. Unlike applets they do not require support for java in the web browser. Unlike CGI, servlets don't use multiple processes to handle separate request. Servers can be handled by separate threads within the same process. Servlets are also portable and platform independent.

A web server is the combination of computer and the program installed on it. Web server interacts with the client through a web browser. It delivers the web pages to the client and to an application by using the web browser and he HTTP protocols respectively.

The define the web server as the package of large number of programs installed on a computer connected to Internet or intranet for downloading the requested files using File Transfer

Protocol, serving e-mail and building and publishing web pages. A web server works on a client server model.

5.5 Conclusion

JSP and Servlets are gaining rapid acceptance as means to provide dynamic content on the Internet. With full access to the Java platform, running from the server in a secure manner, the application possibilities are almost limitless. When JSPs are used with Enterprise JavaBeans technology, e-commerce and database resources can be further enhanced to meet an enterprise's needs for web applications providing secure transactions in an open platform. J2EE technology makes it easy to develop, deploy and use web server applications instead of mingling with other technologies such as CGI and ASP. There are many tools for facilitating quick web software development and to easily convert existing server-side technologies to JSP and Servlets.

CHAPTER 6
IMPLEMENTATION

CHAPTER 6

IMPLEMENTATION

6.1 GENERAL

This chapter describes the implementation of searched based application. It deals with the source code for main viewpoint for Anonymous Database Management.

Index.jsp;

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
    <link rel="stylesheet" href="css/bootstrap.min.css">
        <script type="text/javascript" src="js/jquery.min.js"></script>
        <script type="text/javascript" src="js/bootstrap.min.js"></script>
        <link rel="stylesheet" href="css/font-awesome.min.css">
        <link rel="stylesheet" href="css/font-awesome.css">
</head>
<style>
body{
background-image:url("image/t1.jpg");
background-size:1375px 650px;
}
ul li{
display:inline;
```

```css
text-decoration: none;
background:RAL 6027;
margin-left:10px;
padding:15px;
 font-size:25px;
}
a{
text-decoration:none;
font-size:25px;
font-family:Bernard MT Condensed;
margin:10px;
color:red;

}
i{
color:pink;
}
</style>
<body>
<ul>
<li><i class="fa fa-home">Home</i></li>
<li><i class="fa fa-user"><a href="clientlog.jsp">CLIENT</a></i></li>
<li><i class="fa fa-building"><a href="officelogin.jsp">OFFICE</a></i></li>
<!-- <li><a href="#.jsp">PICKER BRANCH</a></li>
<li><a href="speciallist.jsp">DELIVER BRANCH</a></li>
<li><a href="speciallist.jsp">HEAD OFFICE</a></li> -->
</ul>
</body>

</html>
```

Clientlogin.jsp;

```jsp
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>


<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
 <link rel="stylesheet" href="css/bootstrap.min.css">


    <script type="text/javascript" src="js/jquery.min.js"></script>
    <script type="text/javascript" src="js/bootstrap.min.js"></script>
     <link rel="stylesheet" href="css/font-awesome.min.css">
      <link rel="stylesheet" href="css/font-awesome.css">
</head>
<style>
body{
background-image:url("image/t3.png");
  background-repeat:no-repeat;
   background-size:40% 180%;
}
.myDiv {
   border: 5px outset #989084;
   background-color: #46556fe6;
 border-radius: 10px;
 width:400px;
 height:250px;
 margin: auto;
```

```css
  padding-top:30px;
  /* box-shadow: 25px 20px 20px #888888; */
 }
.myDiv2 {
  font-size:25px;
  font-style: italic;
font-weight: bold;
color:#331100;
}
a{
text-decoration:none;
color:black;}
span{
color:red;
}
.myDiv2{
color:black;
font-variant: small-caps;
}
</style>
</head>
<body>
<center>
<div class="myDiv2">
<span>L</span>ogin <span>P</span>age
</div>
</center>
<br><br><br><br><br><br>
<form action="clientlog" method="post">
<center>
```

```html
<i class="fa fa-user fa-x"><input type="email" name="email" placeholder="Enter  Name"
style="width:280px;height:40px;border-radius: 10px;text-align:center;"></i><br><br>
  <i class="fa fa-key fa-x"><input type="password" name="pass" placeholder="Password"
style="width:280px;height:40px;border-radius: 10px;text-align:center;"></i><br><br>
  <input type="submit" value="Submit" style="width:100px;height:40px;border-radius:
10px;"><br><br>
<a href="clientreg.jsp"><span>NEW </span>USER <span>REGISTER </span>HERE</a>
 </center>
 </form>
 </body>
</html>
```

Clientreg.jsp;

```jsp
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>


<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
  <link rel="stylesheet" href="css/bootstrap.min.css">


    <script type="text/javascript" src="js/jquery.min.js"></script>
    <script type="text/javascript" src="js/bootstrap.min.js"></script>
    <link rel="stylesheet" href="css/font-awesome.min.css">
    <link rel="stylesheet" href="css/font-awesome.css">
</head>
<style>
body{
background-image:url("image/t4.jpg");
```

```css
    background-repeat:no-repeat;
    background-size:cover;
}
.myDiv {
    border: 5px outset #989084;
    background-color: #46556fe6;
  border-radius: 10px;
  width:400px;
  height:250px;
  margin: auto;
  padding-top:30px;
  /* box-shadow: 25px 20px 20px #888888; */


}

.myDiv2 {
  font-size:25px;
  font-style: italic;
font-weight: bold;
color:#331100;
}
a{
text-decoration:none;
color:black;}
span{
color:red;


}
.myDiv2{
color:white;
font-variant: small-caps;
```

```
}
i{
font-size:20px;
color:skyblue;
margin:10px;
}

</style>
</head>
<body>
<center>
<div class="myDiv2">
 <span>R</span>egister <span>P</span>age
</div>
</center>
<br><br><br><br><br><br>
<form action="clientreg" method="post">
<center>
<i class="fa fa-user fa-1x"><input type="text" name="name" placeholder="ENTER  NAME"
style="width:280px;height:40px;border-radius: 10px;text-align:center;"></i><br><br>
<i class="fa fa-envelope-square fa-1x"> <input type="email" name="email"
placeholder="ENTER EMAIL" style="width:280px;height:40px;border-radius: 10px;text-
align:center;"></i><br><br>
  <i class="fa fa-key fa-1x"><input type="password" name="pass" placeholder="Password"
style="width:280px;height:40px;border-radius: 10px;text-align:center;"></i><br><br>
  <i class="fa fa-key fa-1x"><input type="password" name="cpass" placeholder="Con
Password" style="width:280px;height:40px;border-radius: 10px;text-
align:center;"></i><br><br>
  <input type="submit" value="Submit" style="width:100px;height:40px;border-radius:
10px;"><br><br>
</div>
```

```
</center>
</form>
</body>
</html>

Clentmain.jsp;

<!DOCTYPE html>
<html>
<head>
<style>
body {
 background-image:url("image/t15.jpg");
 background-repeat:no-repeat;
 background-size:cover;
 margin: 0;
}
ul {
 list-style-type: none;
 margin: 0;
 padding: 0;
 width: 15%;
 background-color: #f1f1f1;
 position: fixed;
 height: 100%;
 overflow: auto;
}

li a {
 display: block;
 color: #000;
 padding: 8px 16px;
```

```
    text-decoration: none;
}

li a.active {
  background-color: #1955a0;
  color: white;
}

li a:hover:not(.active) {
  background-color: #92a8d1;
  color: white;
}
</style>
</head>
<body>
<ul>
  <li><a class="active" href="#home">HOME</a></li>
   <li><a href="form.jsp">FORM</a></li>
 <!-- <li><a href="deliverystatus.jsp">Delivery Status</a></li> -->
   <li><a href="deliverytime.jsp">DELIVERY</a></li>
  <li><a href="feedback.jsp">FEEDBACK</a></li>
  <li><a href="index.jsp">LOGOUT</a></li>
</ul>
<div style="margin-right:50%;height:10px;padding: 1px 91px;
}">
</div>
</body>
</html>
```

Pickerreg.jsp;

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
```

```
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<style>
body {
    background: url(image/t7.jpeg)no-repeat 0px 0px;

    background-size: 100% 100%;
    min-height: 795px;
        position:relative;
.myDiv {
  border: 5px outset #77c732;
  background-color: lightblue;
  border-radius: 10px;
  width:500px;
  height:330px;
  margin: auto;
  padding-top:30px;
  box-shadow: 25px 20px 20px #888888;
  }
.myDiv2 {
  font-size:25px;
font-style: italic;
font-weight: bold;
color: #77c732;
color:red;
}
```

```html
.span{

}
</style>
</head>
<body>
<br><br><br><br><br><br><br><br><br><br>
<center>
</center>
<br><br>
<div class="myDiv">
<center>
 <form action="pickerregac.jsp" method="post">
 <input type="text" name="username" placeholder="STAFFNAME"
style="width:280px;height:40px;border-radius: 10px;text-align:center;"><br><br>


     <select id="appointment_for" name="fdistrict" required
style="width:280px;height:40px;border-radius: 10px;text-align:center;">
     <option value="CHENNAI">CHENNAI</option>
     <option value="TRICY">TRICY</option>
     <option value="KANYAKUMARI">KANYAKUMARI</option>
     <option value="COIMBATORE">COIMBATORE</option>
     <option value="THIRUNELVELI">THIRUNELVELI</option>
     <option value="ERODE">ERODE</option>
</select><br><br>
 <input type="password" name="pass" placeholder="Password" id="password1"
style="width:280px;height:40px;border-radius: 10px;text-align:center;"><br><br>
 <input type="password" name="cpass" placeholder="Confirm Password" id="password2"
style="width:280px;height:40px;border-radius: 10px;text-align:center;"><br><br>
 <input type="submit" value="Submit" style="width:100px;height:40px;border-radius:
10px;"><br><br>
```

```
    </form>
    </center>
</div>
</body>
<script>
window.onload = function () {
document.getElementById("password1").onchange = validatePassword;
document.getElementById("password2").onchange = validatePassword;
                            }
function validatePassword() {
var pass2 = document.getElementById("password2").value;
var pass1 = document.getElementById("password1").value;
if (pass1 != pass2)
document.getElementById("password2").setCustomValidity("Password Doesn't Match");
else
document.getElementById("password2").setCustomValidity(");
                            //empty string means no validation error
                            }
</script>
</html>
```

Pickermain.jsp;

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
    <link rel="stylesheet" href="css/bootstrap.min.css">
```

```html
<script type="text/javascript" src="js/jquery.min.js"></script>
<script type="text/javascript" src="js/bootstrap.min.js"></script>
<link rel="stylesheet" href="css/font-awesome.min.css">
<link rel="stylesheet" href="css/font-awesome.css">
</head>
<style>
body{
background-image:url("image/t18.jpg");
background-size:1375px 650px;
}
ul li{
display:inline;

text-decoration: none;
background:RAL 6027;
margin-left:10px;
padding:15px;
font-size:25px;
}
a{
text-decoration:none;
font-size:25px;
font-family:Bernard MT Condensed;
margin:10px;
color:green;

}
i{
color:green;
}
```

```
</style>
<body>
<ul>
<li><i class="fa fa-home">Home</i></li>
<li><i class="fa fa-user"><a href="parchalview.jsp">VIEW </a></i></li>
<li><i class="fa fa-building"><a href="officemain.jsp">Logout</a></i></li>
<!-- <li><a href="#.jsp">PICKER BRANCH</a></li>
<li><a href="speciallist.jsp">DELIVER BRANCH</a></li>
<li><a href="speciallist.jsp">HEAD OFFICE</a></li> -->
</ul>
</body>
</html>
```

Headlog.jsp;

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<style>
body{
background-image:url("image/t19.jpg");
background-size: 100%;
}
.myDiv {
  background-color: #ab766b17;
```

```
    border-radius: 10px;
    width:400px;
    height:250px;
    margin: auto;
    padding-top:30px;
    /* box-shadow: 25px 20px 20px #888888; */
}
.myDiv2 {
    font-size:25px;
    font-family: Castellar;
color:#331100;
}
</style>
</head>
<body>
<br><br><br>
<center>
<div class="myDiv2">

HEAD OFFICE
</div>
</center>
<br><br><br><br><br><br>
<center>
<div class="myDiv">
 <form action="headlogAc.jsp" method="post">
<input type="email" name="email" placeholder="email"
style="width:280px;height:40px;border-radius: 10px;text-align:center;     background-color:
#cea966f2;"><br><br>
```

```html
  <input type="password" name="pass" placeholder="Password"
style="width:280px;height:40px;border-radius: 10px;text-align:center; background-color:
#cea966f2;"><br><br>
  <input type="submit" value="Submit" style="width:100px;height:40px;border-radius: 10px;
background-color: #cea966f2;"><br><br>
 </form>
  </div>
</center>
</body>
</html>
```

Clientlog.java;

```java
package servlet;

import imple.imple;

import inter.inter;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import bean.
```
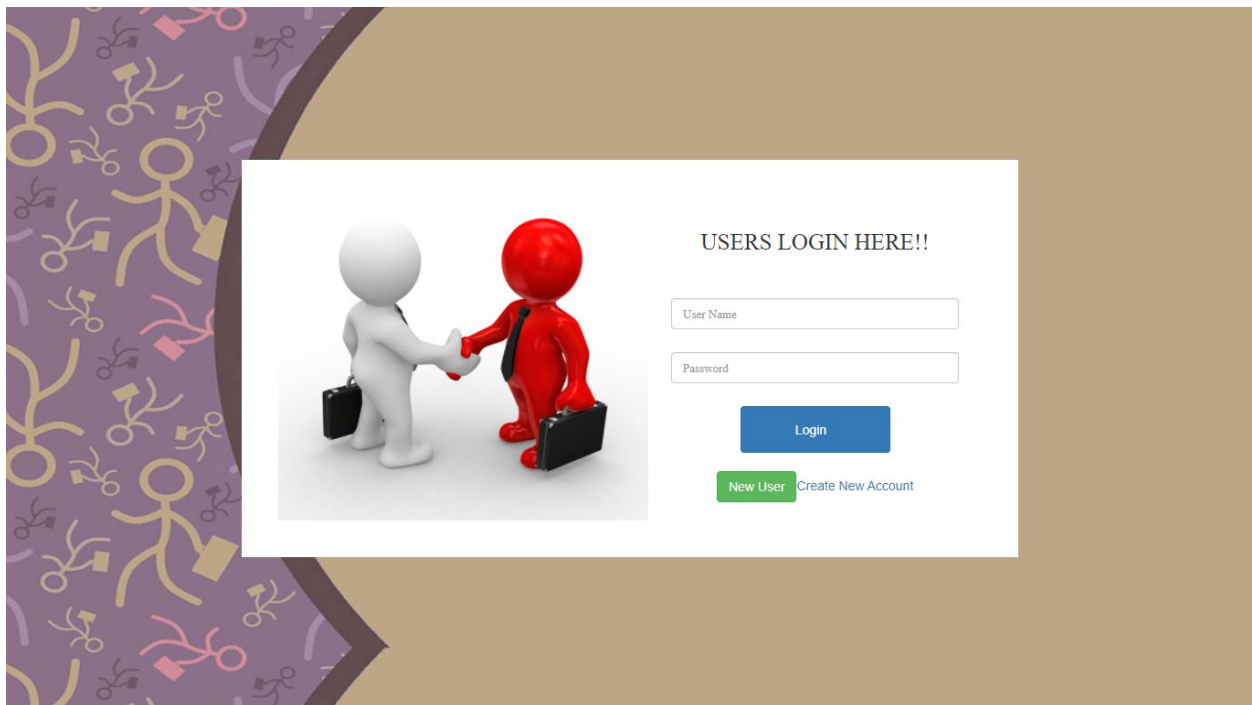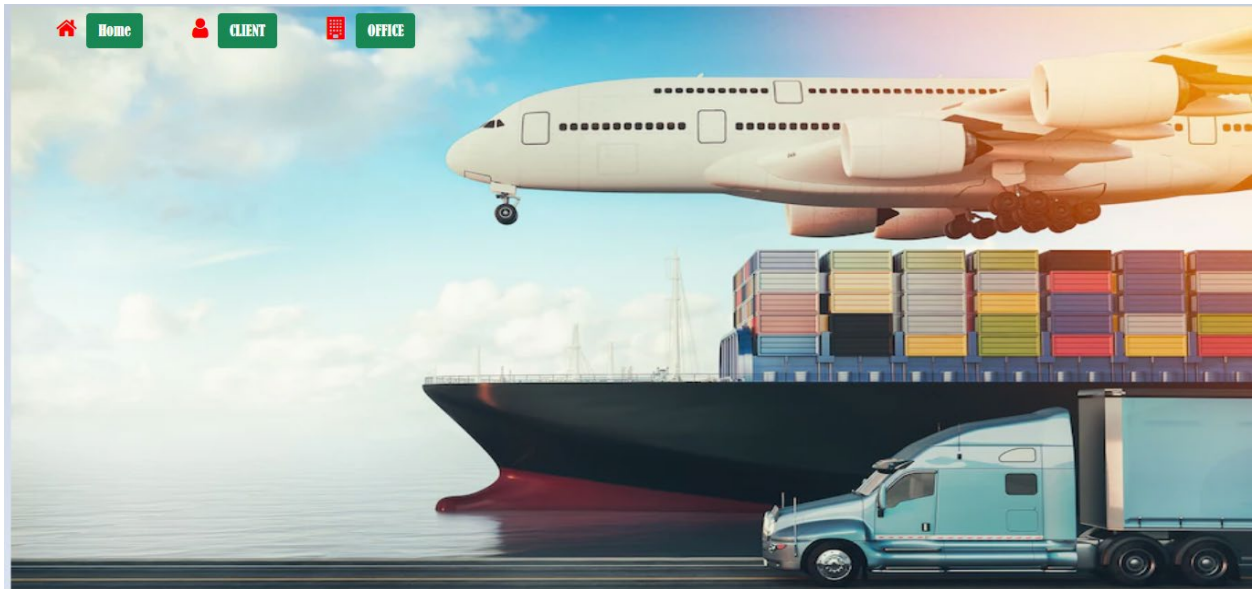
CHAPTER 7
SNAPSHOTS

CHAPTER 7

SNAPSHOTS

Client Register:

CHAPTER 8
SOFTWARE TESTING

CHAPTER 8

SOFTWARE TESTING

8.1. FEASIBILITY STUDY

Feasibility studies aim to uncover the strengths and weaknesses of the existing business or proposed venture, opportunities and threats as presented objectively and rationally by the environment, the resources required to carry through, and ultimately the prospects for success. They are 3 types of feasibility:

• Economical feasibility

• Technical feasibility

• Operational feasibility

8.1.1. ECONOMICAL FEASIBILITY

The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. to estimate whether the new system will perform adequately or not.

8.1.2. TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources.

8.1.3. OPERATIONAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

8.2. SYSTEM TESTING

The software, which has been developed, has to be tested to prove its validity. Testing is the least creative phase of the whole cycle of system design. In the real sense it is the phase, which helps to bring out the creativity of the other phases makes it shine.

8.2.1. VARIOUS LEVELS OF TESTING

1. White Box Testing

2. Black Box Testing

3. Unit Testing

4. Functional Testing

5. Performance Testing

6. Integration Testing

7. Validation Testing

8. System Testing

9. Output Testing

10. User Acceptance Testing

8.2.1.1. WHITE BOX TESTING

White-box testing, sometimes called glass-box, is a test case design method that uses the control structure of the procedural design to derive test cases. Using White Box testing methods, we can derive test cases that:

• Guarantee that all independent paths within a module have been exercised at least once

• Exercise all logical decisions on their true and false sides.

• Execute all loops at their boundaries and within their operational bounds.

• Exercise internal data structures to assure their validity.

8.2.1.2. BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. You cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

In this testing by knowing the internal operation of a product, test can be conducted to ensure that "all gears mesh", that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

8.2.1.3. UNIT TESTING

Unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method.

Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process.

Unit testing is software verification and validation method in which the individual units of source code are tested fit for use. A unit is the smallest testable part of an application. In this testing, each class is tested to be working satisfactorily.

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration.

8.2.1.4. FUNCTIONAL TESTING

Functional testing is a quality assurance (QA) process and a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (not like in white-box testing). Functional Testing usually describes what the system does. Functional testing differs from system testing in that functional testing "verifies a program by checking it against ... design document(s) or specification(s)", while system testing "validate a program by checking it against the published user or system requirements" (Kane, Falk, Nguyen 1999, p. 52). Functional testing typically involves five steps. The identification of functions that the software is expected to perform,

1. The creation of input data based on the function's specifications

2. The determination of output based on the function's specifications

3. The execution of the test case

4. The comparison of actual and expected outputs.

8.2.1.5. PERFORMANCE TESTING

In general testing performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate, measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage.

Performance testing is a subset of performance engineering, an emerging computer science practice which strives to build performance into the implementation, design and architecture of a system.

8.2.1.6. INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when put together. The problem of course, is "putting them together"- interfacing. There may be the chances of data lost across on another's sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready. All the errors found in the system are corrected for the next phase.

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e., assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested, and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly

for example across procedure calls or process activations, and this is done after testing individual modules, i.e., unit testing.

8.2.1.7. VALIDATION TESTING

Verification and Validation are independent procedures that are used together for checking that a product, service, or system meets requirements and specifications and that it full fills its intended purpose. These are critical components of a quality management system such as ISO 9000. The words "verification" and "validation" are sometimes preceded with "Independent" (or IV&V), indicating that the verification and validation is to be performed by a disinterested third party.

It is sometimes said that validation can be expressed by the query "Are you building the right thing?" and verification by "Are you building it right?". In practice, the usage of these terms varies. Sometimes they are even used interchangeably.

8.2.1.8. SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called *assemblages*) or between any of the *assemblages* and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification.

8.2.1.9. OUTPUT TESTING

After performing the validation testing, next step is output testing of the proposed system since no system could be useful if it does not produce the required output generated or considered in to two ways. One is on screen, and another is printed format. The output comes as the specified requirements by the user. Hence output testing does not result in any correction in the system.

8.2.1.10. USER ACCEPTANCE TESTING

User acceptance of a system is the factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required.

• Input screen design.

• Output screen design.

• Inline message to guide user.

• Format of the ad-hoc reports and other outputs.

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using the test data. While testing the system by using test data errors are again uncovered and correct.

CHAPTER 9

FUTURE ENHANCEMENT

CHAPTER 9
FUTIRE ENHANCEMENTS

1. Implementing a genuine data set framework.
2. Improving the productivity of conventions, as far as number of messages traded and concerning their sizes, also.
3. Implement using two are more algorithms.

CHAPTER 10
CONCLUSION

CHAPTER 10
CONCLUSION

We have proposed for constant provisioning in cloud-based modern applications. It pointed toward getting a streamlined execution cost progressively with the thought of client characterized security arrangement constraints.in this cycle picker center receive the client parcel and send conveyance address everything is checking in lastly branch supervisor send the day to day picking parcel and conveyance in branch center point each detail to send the administrative center. The branch supervisor report will be sent secure the administrative center. Report will be encoded first utilizing AES Algorithm.

REFERENCE OR BIBLIOGRAPHY:

[1] S. Zhang, X. Li, Z. Tan, T. Peng, and G. Wang, "A caching and spatial K-anonymity driven privacy enhancement scheme in continuous locationbased services," Future Gener. Comput. Syst., vol. 94, pp. 40–50, 2019.

[2] Q. S. Hua et al., "Faster parallel core maintenance algorithms in dynamic graphs," IEEE Trans. Parallel Distrib. Syst., vol. 31, no. 6, pp. 1287–1300, Jun. 2020.

[3] A. Jolfaei, X.-W. Wu, and V. Muthukkumarasamy, "On the security of permutation-only image encryption schemes," IEEE Trans. Inf. Forensics Secur., vol. 11, no. 2, pp. 235–246, Feb. 2016.

[4] W. Gong, L. Qi, and Y. Xu, "Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment," Wireless Commun. Mobile Comput., vol. 2018, 2018, Art. no. 3075849.

[5] S. Zhang, G. Wang, M. Z. A. Bhuiyan, and Q. Liu, "A dual privacy preserving scheme in continuous location-based services," IEEE Internet Things J., vol. 5, no. 5, pp. 4191–4200, Oct. 2018.

[6] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, "BeCome: Blockchain-enabled computation offloading for IoT in mobile edge computing," IEEE Trans. Ind. Informat., vol. 16, no. 6, pp. 4187–4195, Jun. 2019.

[7] S. Wang, L. Hu, Y. Wang, Q. Z. Sheng, M. Orgun, and L. Cao, "Modeling multi-purpose sessions for next-item recommendations via mixturechannel purpose routing networks," in Proc. 28th Int. Joint Conf. Artif. Intell., 2019, pp. 1–7.

8] B. Zhao, Y. Wang, Y. Li, Y. Gao, and X. Tong, "Task allocation model based on worker friend relationship for mobile crowdsourcing," Sensors, vol. 19, no. 4, p. 921, 2019.

[9] S. Meng, Q. Li, T. Wu, W. Huang, J. Zhang, and W. Li, "A fault-tolerant dynamic scheduling method on hierarchical mobile edge cloud computing," Comput. Intell., vol. 35, no. 3, pp. 577–598, 2019.

[10] Y. Wang, Z. Cai, Z. H. Zhan, Y.-J. Gong, and X. Tong, "An optimization and auction-based incentive mechanism to maximize social welfare for mobile crowdsourcing," IEEE Trans. Comput. Social Syst., vol. 6, no. 3, pp. 414–429, Jun. 2019.

[11] D. Yu et al., "Implementing abstract MAC layer in dynamic networks," IEEE Trans. Mobile Comput., to be published. doi: 10.1109/TMC.2020.2971599.

[12] S. Zhang, K. K. R. Choo, Liu Q, and G. Wang, "Enhancing privacy through uniform grid and caching in location-based services," Future Gener. Comput. Syst., vol. 86, pp. 881–892, 2018.

[13] H. Liu, H. Kou, C. Yan, and L. Qi, "Link prediction in paper citation network to construct paper correlation graph," EURASIP J. Wireless Commun. Netw., vol. 2019, no. 1, pp. 1–12, 2019.

[14] Y. Wang, Z. Cai, X. Tong, Y. Gao, and G. Yin, "Truthful incentive mechanism with location privacy-preserving for mobile crowdsourcing systems," Comput. Netw., vol. 135, pp. 32–43, 2018.

[15] A. Jolfaei and K. Kant, "A lightweight integrity protection scheme for low latency smart grid applications," Comput. Secur., vol. 86, pp. 471–483, 2019

[16] L. Qi, W. Dou, W. Wang, G. Li, H. Yu, and S. Wan, "Dynamic mobile crowdsourcing selection for electricity load forecasting," IEEE Access, vol. 6, pp. 46926–46937, 2018.

[17] A. Jolfaei, X.-W. Wu, and V. Muthukkumarasamy, "Comments on the security of 'diffusion–substitution based gray image encryption' scheme," Digit. Signal Process., vol. 32, pp. 34–36, 2014.

[18] S. Abrishami, M. Naghibzadeh, and D. H. J. Epema, "Deadlineconstrained workflow scheduling algorithms for infrastructure as a service clouds," Future Gener. Comput. Syst., vol. 29, no. 1, pp. 158–169, 2013.

[19] X. Xu, X. Zhang, M. Khan, W. Dou, S. Xue, and S. Yu, "A balanced virtual machine scheduling method for energy-performance trade-offs in cyber-physical cloud systems," Future Gener. Comput. Syst., vol. 105, pp. 789–799, 2017.

[20] F. Li, D. Yu, H. Yang, J. Yu, H. Karl, and X. Cheng, "Multi-armed-banditbased spectrum scheduling algorithms in wireless networks: A survey," IEEE Wireless Commun., vol. 27, no. 1, pp. 24–30, Feb. 2020.

[21] H. R. Boveiri, R. Khayami, M. Elhoseny, and M. Gunasekaran, "An efficient swarm-intelligence approach for task scheduling in cloud-based internet of things applications," J. Ambient Intell. Humanized Comput., vol. 10, no. 9, pp. 3469–3479, 2019.

[22] Q. Zhang, M. Lin, L. T. Yang, Z. Chen, S. U. Khan, and P. Li, "A double deep Q-learning model for energy-efficient edge scheduling," IEEE Trans. Services Comput., vol. 12, no. 5, pp. 739–749, Sep./Oct. 2019.

[23] F. Juarez, J. Ejarque, and R. M. Badia, "Dynamic energy-aware scheduling for parallel task-based application in cloud computing," Future Gener. Comput. Syst., vol. 78, pp. 257–271, 2018.

[24] X. Xu, H. Cao, Q. Geng, X. Liu, F. Dai, and C. Wang, "Dynamic resource provisioning for workflow scheduling under uncertainty in edge computing environment," Concurrency Comput.-Pract. Experience, to be published. doi: 10.1002/cpe.5674.

[25] S. Wang, L. Hu, L. Cao, X. Huang, D. Lian, and W. Liu, "Attentionbased transactional context embedding for next-item recommendation," in Proc. 32nd AAAI Conf. Artif. Intell., New Orleans, LA, USA, Feb. 2018, pp. 2532–2539.

[26] V. Singh, I. Gupta, and P. K. Jana, "A novel cost-efficient approach for deadline-constrained workflow scheduling by dynamic provisioning of resources," Future Gener. Comput. Syst., vol. 79, pp. 95–110, 2018.

[27] C. A. Aslam, Y. L. Guan, and K. Cai, "Edge-based dynamic scheduling for belief-propagation decoding of LDPC and RS codes," IEEE Trans. Commun., vol. 65, no. 2, pp. 525–535, Feb. 2017.

[28] X. Li, J. Wan, H. N. Dai, M. Imran, M. Xia, and A. Celesti, "A hybrid computing solution and resource scheduling strategy for edge computing in smart manufacturing," IEEE Trans. Ind. Informat., vol. 15, no. 7, pp. 4225–4234, Jul. 2019.

[29] S. Sharif, P. Watson, J. Taheri, S. Nepal, and A. Y. Zomaya, "Privacy-aware scheduling SaaS in high performance computing environments," IEEE Trans. Parallel Distrib. Syst., vol. 28, no. 4, pp. 1176–1188, Apr. 2017.

[30] Y.Wen, J. Liu,W. Dou, X. Xu, B. Cao, and J. Chen, "Scheduling workflows with privacy protection constraints for big data applications on cloud," Future Gener. Comput. Syst., vol. 108, pp. 1084–1091, 2020.

[31] H. Zhang and K. Zeng, "Pairwise Markov chain: A task scheduling strategy for privacy-preserving SIFT on edge," in Proc. IEEE Conf. Comput. Commun., Apr. 2019, pp. 1432–1440.

[32] H. Sheng et al., "Mining hard samples globally and efficiently for person re-identification," IEEE Internet Things J., to be published, doi: 10.1109/JIOT.2020.2980549.