

A Major Project Report

On

## **Cyber Physical Social Systems Based On Privacy Preserved Profile Publishing**

*Submitted to JNTU HYDERABAD*

*In Partial Fulfilment of the requirements for the Award of Degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

Submitted

By

**Sheetal Patangay (168R1A05H3)**

Under the Esteemed guidance of

**Dr. M. Kumara Swamy**

Professor, Department of CSE



**Department of Computer Science & Engineering**

**CMR ENGINEERING COLLEGE**

(Approved by AICTE, NEW DELHI, Affiliated to JNTU, Hyderabad)

Kandlakoya, Medchal Road, R.R. Dist. Hyderabad-501 401)

**2019-2020**

# **CMR ENGINEERING COLLEGE**

*(Accredited by NBA, Approved by AICTE NEW DELHI, Affiliated to JNTU, Hyderabad)*  
*Kandlakoya, Medchal Road, Hyderabad-501 401*

## **Department of Computer Science & Engineering**



### **CERTIFICATE**

This is to certify that the project entitled "**CYBER PHYSICAL SOCIAL SYSTEMS BASED ON PRIVACY PRESERVED PROFILE PUBLISHING**" is a Bonafide work carried out by

**Sheetal Patangay (168R1A05H3)**

In partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** from CMR Engineering College, affiliated to JNTU, Hyderabad, under our guidance and supervision.

The results presented in this project have been verified and are found to be satisfactory. The results embodied in this project have not been submitted to any other university for the award of any other degree or diploma.

#### **Internal Guide**

#### **Major Project Coordinator**

#### **Head of the Department**

**Dr. M. Kumara Swamy**

**Professor**

Department of CSE

CMREC, Hyderabad

**Mrs.G.Sumalatha**

**Associate Professor**

Department of CSE

CMREC, Hyderabad

**Dr. Sheo Kumar**

**Professor & H.O.D**

Department of CSE

CMREC, Hyderabad

## **DECLARATION**

This is to certify that the work reported in the present project entitled "**CYBER PHYSICAL SOCIAL SYSTEMS BASED ON PRIVACY PRESERVED PROFILE PUBLISHING**" is a record of bonafide work done by us in the Department of Computer Science and Engineering, CMR Engineering College, JNTU Hyderabad. The reports are based on the project work done entirely by us and not copied from any other source. We submit our project for further development by any interested students who share similar interests to improve the project in the future.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

**Sheetal Patangay**

**(168R1A05H3)**

## **ACKNOWLEDGMENT**

I am extremely grateful to **Dr. A. Srinivasula Reddy**, Principal, and **Dr. Sheo Kumar**, HOD, **Department of CSE, CMR Engineering College** for their constant support.

I am extremely thankful to **Dr. M. Kumra Swamy**, Professor, Internal Guide, Department of CSE, for his constant guidance, encouragement, and moral support throughout the project.

I will be failing in our duty if I do not acknowledge with thanks to the authors of thereferences and other literature referred to in this Project.

I thank all staff members and friends for all the help and coordination extended in bringing out this Project successfully in time.

Finally, I am very thankful to my parents who guided me through every step.

**Sheetal Patangay** (168R1A05H3)

# CONTENTS

<b>TOPIC</b>	<b>PAGENO</b>
<b>ABSTRACT</b>	i
<b>List of Figures</b>	ii
<b>1. INTRODUCTION</b>	
1.1 Introduction & Objectives	1
1.2 Purpose of the Project	2
1.3 Existing system & Disadvantages	3
1.4 Proposed system with Features	4
<b>2. LITERATURE SURVEY</b>	6
<b>3. SOFTWARE REQUIREMENT ANALYSIS</b>	
3.1 Problem Specification	7
3.2 Modules and their Functionalities	7
3.3 Feasibility Study	8
<b>4. SOFTWARE &amp; HARDWARE REQUIREMENTS</b>	
4.1 Software requirements	10
4.2 Hardware requirements	10
<b>5. SOFTWARE DESIGN</b>	
5.1 Data Flow Diagrams	11
5.2. UML Diagram	11
<b>6. SOFTWARE ENVIRONMENT</b>	17
<b>7. CODING AND IMPLEMENTATION</b>	
7.1 Sample code	33
<b>8. SYSTEM TESTING</b>	
8.1 Testing Strategies	44
8.2 Testing Design	48
<b>9. OUTPUT SCREENS</b>	50
<b>10. CONCLUSION</b>	63
<b>11. REFERENCES</b>	65

## ABSTRACT

Due to the close correlation with individual's physical features and status, the adoption of Cyber-Physical Social Systems (CPSSs) has been inevitably hindered by users' privacy concerns. Such concerns keep growing as our mobile devices have more embedded sensors, while the existing countermeasures only provide incapable and limited privacy preservation for sensitive physical information. We formulate both the privacy concerns and user expectations in CPSSs based on real-world knowledge. We also design a corresponding data publishing mechanism for users. It regulates the publishing behaviors to hide sensitive physical profiles. Meanwhile, the published data retain comprehensive social profiles for users. Our analysis demonstrates that the mechanism achieves a local maximized performance on the aspect published data size. The experiment results towards real datasets reveal that the performance is comparable to the global optimal one.

## **LIST OF FIGURES**

<b>S.NO</b>	<b>DESCRIPTION</b>	<b>PAGENO</b>
5.1	Data Flow Diagram	11
5.2	UML Diagrams	11
5.2.1	Use Case Diagram	12
5.2.2	Sequence Diagram	13
5.2.3	Class Diagram	14
5.2.4.1	Activity Diagram of Admin	15
5.2.4.2	Activity Diagram of User	16

# 1. INTRODUCTION

## 1.1 Introduction

Cyber-Physical Social Systems (CPSSs), as a new extension of social networks, is changing our life according to the recent studies. Users update their physical data collected through pervasive sensors in their mobile devices. They even act as "sensors" themselves by taking photos or making comments. In such systems, the users are the creators and builders. They participate in both the computation and formation of the system. On the other hand, users establish their profiles as they do in a regular social network by publishing their sensed data. They form their own reputations and achieve a self-actualization. A typical CPSS is the Local Business Service System (LBSS), where users initially visit some Point of Interests (PoIs) in their cities and upload photos and comments like "intellectual sensors". Others who are in favor of these comments would praise or follow them. Unfortunately, while enhancing the functionality of existing Cyber Physical Systems, the CPSS also brings privacy threats to users, since the generated data usually reveal some private information such as locations, motions, and personal habits. The users could suffer physical threats, which are far more harmful than advertisements or spam mails. Therefore, the users face severe challenges when sharing their data in CPSSs. As a solution, this paper studies the privacy preserved data publishing problem in CPSSs. The privacy issues in CPSSs are actually different from the ones in regular social networks or the ones towards physical data. Most previous works focusing on data privacy issues only preserve privacy for single or several records. Take the privacy preservation for geographical locations as an example. The corresponding problems are well-studied ranging from hiding sensitive locations to avoiding inferring locations from public records. The typical techniques include true location obfuscation, anonymity, etc. However, these countermeasures are incapable for CPSS since users often contribute long-term behaviors and expect their whole physical profiles or patterns to be privacy preserved.

More specifically, as in a social network, users establish their profiles via long-term participation. They publish and share numerous records revealing their general physical profiles such as mobility pattern in daily life. The physical profiles are closely related to their behaviors in the physical world. For example, the mobility pattern refers to a user's moving regularity, like appearance possibility in an urban area, visit of abnormal POIs and many more. Adversaries can crawl the published records from the systems and infer the profile to obtain the private information such as working and

residential areas, abnormal visiting behaviors, or even potential changes in one's life. Such information may be further used for advertisement delivery or even physical stalking. According to the study on a real-case dataset, users with only a moderate number of published records already reveal some features of their mobility patterns.

Therefore, there must be a well-designed tool to help users regulate their publishing behaviors in CPSSs. Meanwhile, CPSS users expect to establish and maintain their social profiles while participating in the computation and formation of the cyber-physical systems. For example, in an LBSS like Yelp or Facebook, while users voluntarily build up the reputation system for Poi's, they also expect to attract more followers and receive more supports. Their social profiles can determine how they look like in social networks. Therefore, they show their activeness, taste, or even life styles through their published records including ratings, comments, photos, and the inherent features of the PoIs. Due to the fact that the social profile can be also derived from the published records, the privacy preserved tool must also take the utility in social networks into consideration. For privacy preservation in social networks, the existing works mainly focus on hiding both the sensitive communities and the sensitive information in them. Some works also consider the sensitive information for a single user, like privacy preservation on sensitive attributes, which are not designed for physical data. Given the challenges and drawbacks on both sides, the privacy preservation problem in publishing records and profiles in CPSSs remains unsolved. To mitigate this gap, we propose a novel framework to handle the record publishing problem in CPSSs considering both privacy and social utility of users. More specifically, the proposed framework can

- 1) Preserve physical profiles by carefully selecting the public records;
- 2) retain utility for individuals within the social networks;
- 3) support a user-friendly control so that users can adjust their preference and sensitivity on privacy.

To the best of our knowledge, this is the first work regarding record publishing in CPSSs considering both the privacy on physical profiles and utility on social profiles. More specifically, the privacy preservation mechanism on physical profiles mainly aims to perturb the adversary's knowledge on users' physical status. It perturbs the probability of a user staying in each status in the physical profile, and reduces the confidence in the frequently appeared status. For the utility in social networks, our mechanism maintains the consistency with the original social profile and publishes a maximum

number of records. It guarantees that users can attract the same type of followers and show their activeness. We formulate this problem as an integer non-linear programming problem and propose a heuristic algorithm. We prove this algorithm can achieve a local maximized result towards published records while following both privacy and utility constraints. The experiments on real data sets reveal that our framework outperforms the existing works, and are comparable to the optimal results. Following are our main contributions:

- We propose novel definitions on the utility on social profiles and privacy on physical profiles for CPSSs, which are more reasonable for real world scenarios.
- We formulate the record publishing problem in CPSSs and propose a heuristic algorithm to select the published records.
- We analyze and prove that the heuristic algorithm can satisfy all the constraints while publishing a maximal number of records for each user.
- We also propose an adaptive algorithm to support the publishing of novel records.
- We extensively evaluate the performance of our framework towards the real-world dataset.

## **1.2 Purpose of the project**

Users update their physical data collected through pervasive sensors in their mobile devices. They even act as "sensors" themselves by taking photos or making comments. . In such systems, the users are the creators and builders. They participate in both the computation and formation of the system. On the other hand, users establish their profiles as they do in a regular social network by publishing their sensed data.

They publish and share numerous records revealing their general physical profiles such as mobility pattern in daily life. The physical profiles are closely related to their behaviors in the physical world. , CPSS users expect to establish and maintain their social profiles while participating in the computation and formation of the cyber-physical systems. We formulate this problem as an integer non-linear programming problem and propose a heuristic algorithm.

### **1.3 Existing System and Disadvantages**

Privacy preservation in cyber-physical systems has been attracting the attention from both academic and industrial communities. This issue draws even more attention in the recent years due to the pervasively embedded sensors in mobile devices. Typical studies include exposure of sensitive information, privacy preservation of physical data, and usage or access to sensor data. Location information, as a type of physical data, is closely related to individuals, which attracts even more attention due to the popularity of GPS devices and other localization techniques. It is a kind of major physical data shared by users in CPSSs. The existing techniques like coarsening, faking, k-anonymity, partial publication are proposed to design Location-Privacy Protection Mechanisms (LPPMs). The location obfuscation mechanism is investigated in the existing system to preserve user profiles on mobile patterns. Our study is to build a framework that can both comprehensively preserve users' mobile patterns and maintain their utility in social networks. A crowd sourcing system is another CPSS instance. Location privacy in crowd sourcing systems is considered by several works. However, they mainly preserve user privacy at single or several locations, which is incapable for long-time behaviors in social networks.

The following are the drawbacks present in the existing system:

- There are no efficient techniques in existing system to protect the Track privacy.
- There is no option to Provide Protection on User Track data.

### **1.4 Proposed System with Features**

In the proposed system, the system introduces an LBSS, which is a typical example of CPSS. This type of systems has a close connection with the physical world, i.e., gathering information for Polis, while also maintains an underlying social network where users may interact with each other. Both the terms and definitions proposed in this systems are explained for this real-world case. In a typical LBSS, like Yelp or Trip Advisor, users act as sensors. They voluntarily visit local businesses in their residential areas, write comments, take photos, and upload information to the system. Meanwhile, users expect to attract more followers as in a general social network. Some LBSS service providers mark the outstanding users as elite users to praise their efforts.

The following are the features of the proposed system:

- The system is more efficient due to Divided One-Step Maximum Record Publishing Algorithm.
- Many techniques are involved in the implementation Follow but No Track like ONE TIME REVIEW PUBLISHING ALGORITHM.

## 2. Literature Survey

Privacy preservation in cyber-physical systems has been attracting the attention from both academic and industrial communities. This issue draws even more attention in the recent years due to the pervasively embedded sensors in mobile devices. Typical studies include exposure of sensitive information, privacy preservation of physical data, and usage or access to sensor data. Xu et al, propose an approach based on the virtual reality techniques to bypass face authentication tests and protect user privacy by avoiding the exposure of their real faces. There work mainly studies the domain-specific data privacy, while ignoring the utility in social networks. Location information, as a type of physical data, is closely related to individuals, which attracts even more attention due to the popularity of GPS devices and other localization techniques. Location information is also a kind of major physical data shared by users in CPSSs. The existing techniques like coarsening, faking, k-anonymity, partial publication are proposed to design Location-Privacy Protection Mechanisms (LPPMs).

The location obfuscation mechanism is investigated in to preserve user profiles on mobile patterns. This work is similar with ours, but it simply notifies users regarding the exposure of their physical profiles. Our study is to build a framework that can both comprehensively preserve users mobile patterns and maintain their utility in social networks. A crowdsourcing system is another CPSS instance. Location privacy in crowdsourcing systems is considered by several works. However, they mainly preserve user privacy at single or several locations, which is incapable for long-time behaviors in social networks. For social networks, some works study the privacy preservation for content publishing. Minkus et al, investigate how the posted photos and comments on Facebook and Instagram would expose sensitive information of children. The work in can infer a user's undisclosed information from public data in social networks, and provides some advices on the hiding of undisclosed information. Some studies regard profiles as relational data, and utilize the corresponding techniques like differential privacy to sanitize published data. They generalize, add noise and release data for general purposes. The work in investigates how the existence of communities impacts published data. There are also some work investigating the attack on contents in anonymous data flows. These works are initially designed for social networks, and do not consider physical data. Therefore, they are not proper for CPSSs.

Partially publishing profile and data is another major methodology in privacy preservation, where only a proportion of data is published, such as insensitive attributes or insignificant locations. Shokri et al, propose a mechanism to randomly publish true locations and fake locations. They formulate the location publishing problem as a Stackleberg game and derive a privacy-optimal solution. The work in predicts a driver's destination according to the current path and historical.

### **3. SOFTWARE REQUIREMENT ANALYSIS**

#### **3.1 Problem specification**

Requirement analysis is a software engineering task that bridges the gap between system software allocation and software design. It provides system engineer to specify software function and performance indicate software's interface with other system elements and establish constraints that software must need. The basis is to obtain a clear picture of the needs and requirements of the end user and also the organization. Analysis involves interaction between analysts and clients. The analyst Research the problem from the question asks the reading existing documents. During analysis it is essential that a complete and consistent set of specification emerge for the system. The essential final consistent specifications are divided into five parts.

1. Problem recognition
2. Evaluating and synthesis
3. Modeling
4. Specification
5. Review

#### **3.2 Modules and their Functionalities**

It has the following modules.

##### **1. Admin**

In this module, the Admin has to login by using valid user name and password. After login successful he can do some operations such as --- Login, View all users and authorize, give click option to view all users locations in GMap using Multiple Markers, View all Friend Request and Response, View all users Visited places details and give click option to view all users visited locations in GMap using Multiple Markers, View all users hidden sensitive information and Visited places details to their Friends, View all users shared photos with comments and ranking, ratings, List no. of users hidden sensitive info and no. of users hidden visited places and give link to show in Chart, List no. of users visited same place and give link to show in Chart, List no. of users located same place and give link to show in Chart, View Photos ranks in chart.

##### **2. User**

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like Register with Location, Sensitive Info(Blood Group, Spouse Name, Company Name, Current Living Location) and Login and View your profiles ,Search Friend Track and Find Friend Request and give privileges (while accepting) like to access Sensitive info, Visited Locations ,access on photos, View all Your Friends Tracked and give click option to view all users locations in GMap using Multiple Markers

show Route path using GMAP, Set All your Visited Locations using GMap lat and long with famous place details like place name, place desc(enc), Specialty, Address, Timings, visited Data and Time, View all Your friends visited locations details with famous place details like place name, place desc(enc), Specialty, Address, Timings and give click option to view all users visited locations in GMap using Multiple Markers, Share your photos to your friends with photo name, photo desc, photo uses, add photo image, View all your shared photos with comments, rank and rating, Views all your friends shared photos(Give rank while viewing) and give comment, Rating.

### **3.3 Feasibility Study**

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. The different feasibilities that have to be analyzed are

- OPERATIONAL FEASIBILITY
- ECONOMIC FEASIBILITY
- TECHNICAL FEASIBILITY

#### **OPERATIONAL FEASIBILITY**

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the Admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

#### **ECONOMIC FEASIBILITY**

Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer based project. As hardware was installed from the beginning & for lots of purposes thus the cost on project of hardware is low. Since the system is a network based, any number of employees connected to the LAN within that organization can use this tool from at any time. The Virtual Private Network is to be developed using the existing resources of the organization. So the project is economically feasible.

#### **TECHNICAL FEASIBILITY**

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The organization needs IBM compatible machines with a graphical web browser connected to the Internet and Intranet. The system is developed for platform Independent environment. The technical feasibility has been carried out. The system is technically feasible for development and can be developed with the existing facility.

## **4. Software and Hardware Requirements**

### **4.1 Software Requirements**

- Operating system : Windows XP/10.
- Coding Language : Java 14.
- Back-end : MYSQL.
- Server : Tomcat Server.

### **4.2 Hardware Requirements**

- RAM : RAM 4GB
- Configuration : i3,i5,i7
- Hard Disk Space : 500 GB
- Processor Speed : 2.00 GHz

## 5. Software Design

### 5.1 Data Flow Diagrams

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

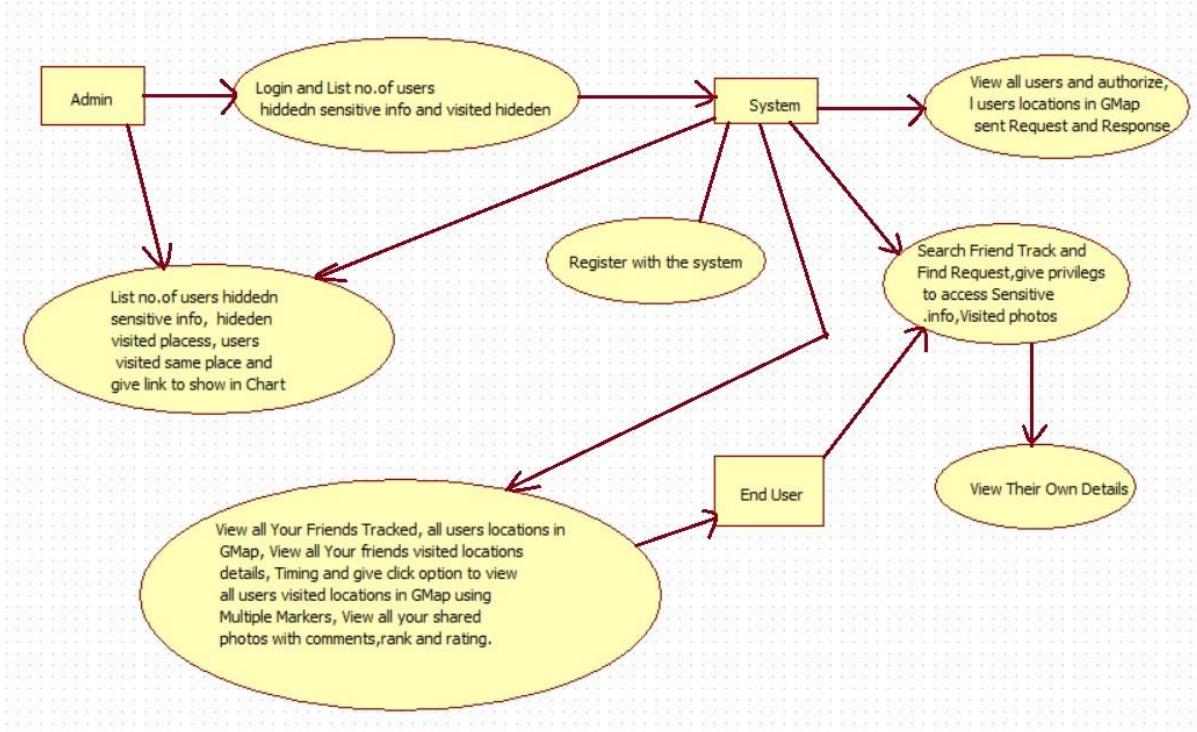


Fig.1. Data-flow Diagram.

### 5.2 UML Diagrams

#### 5.2.1 Use Case Diagram

A use case in a use case diagram is a visual representation of distinct business functionality in a system. The key term here is "distinct business functionality." To choose a business process as a likely candidate for modeling as a use case, you need to ensure that the business process is discrete in nature.

As the first step in identifying use cases, you should list the discrete business functions in your problem statement. Each of these business functions can be classified as a potential use case. Remember that identifying use cases is a discovery rather than a creation. As business

functionality becomes clearer, the underlying use cases become more easily evident. A use case is shown as an ellipse in a use case diagram.

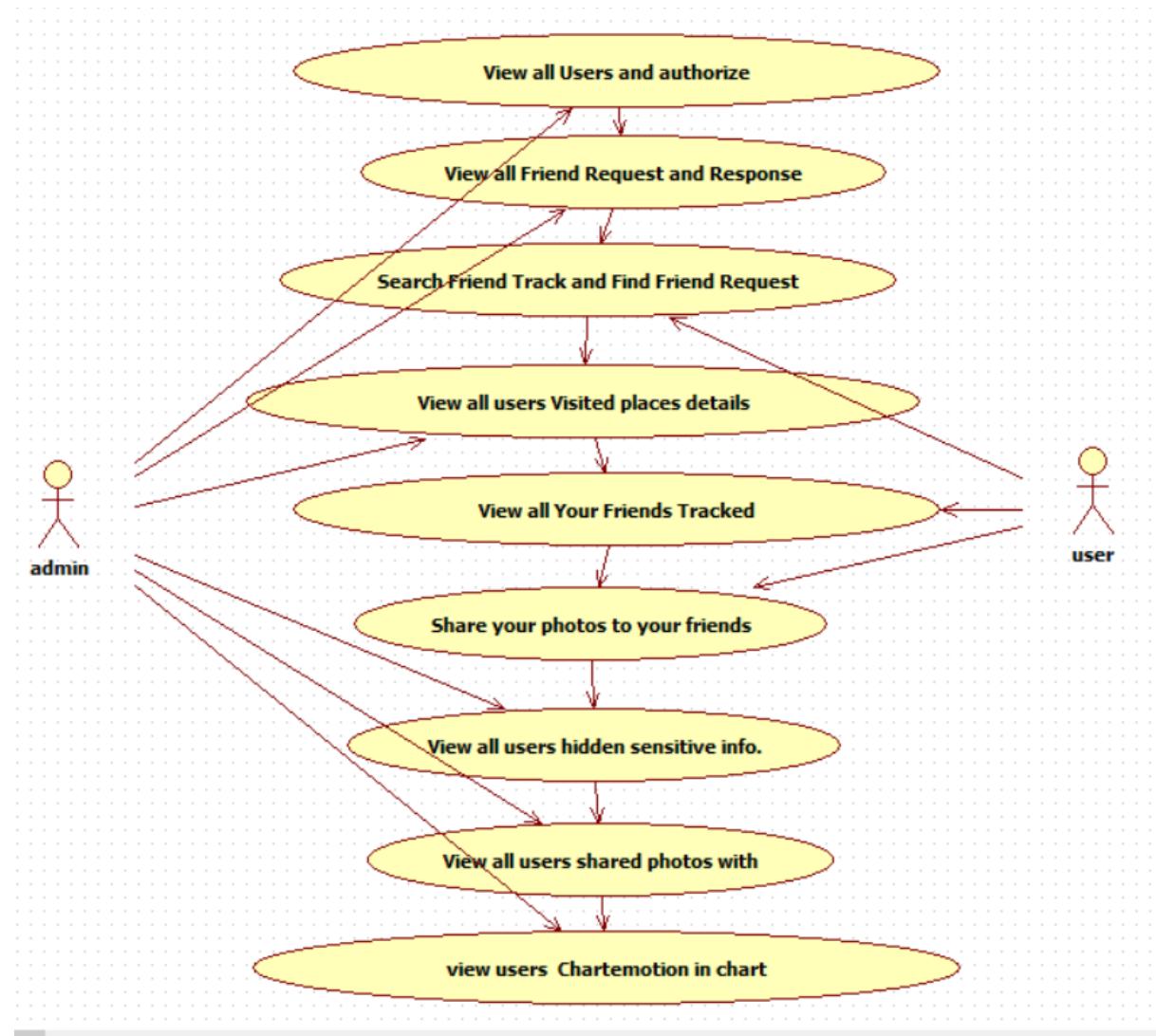


Fig.2. Use Case Diagram.

### 5.2.2 Sequence Diagram

UML sequence diagrams are used to represent the flow of messages, events and actions between the objects or components of a system. Time is represented in the vertical direction showing the sequence of interactions of the header elements, which are displayed horizontally at the top of the diagram.

Sequence Diagrams are used primarily to design, document and validate the architecture, interfaces and logic of the system by describing the sequence of actions that need to be performed to complete a task or scenario. UML sequence diagrams are useful design tools because they provide a dynamic view of the system behavior which can be difficult to extract from static diagrams or specifications.

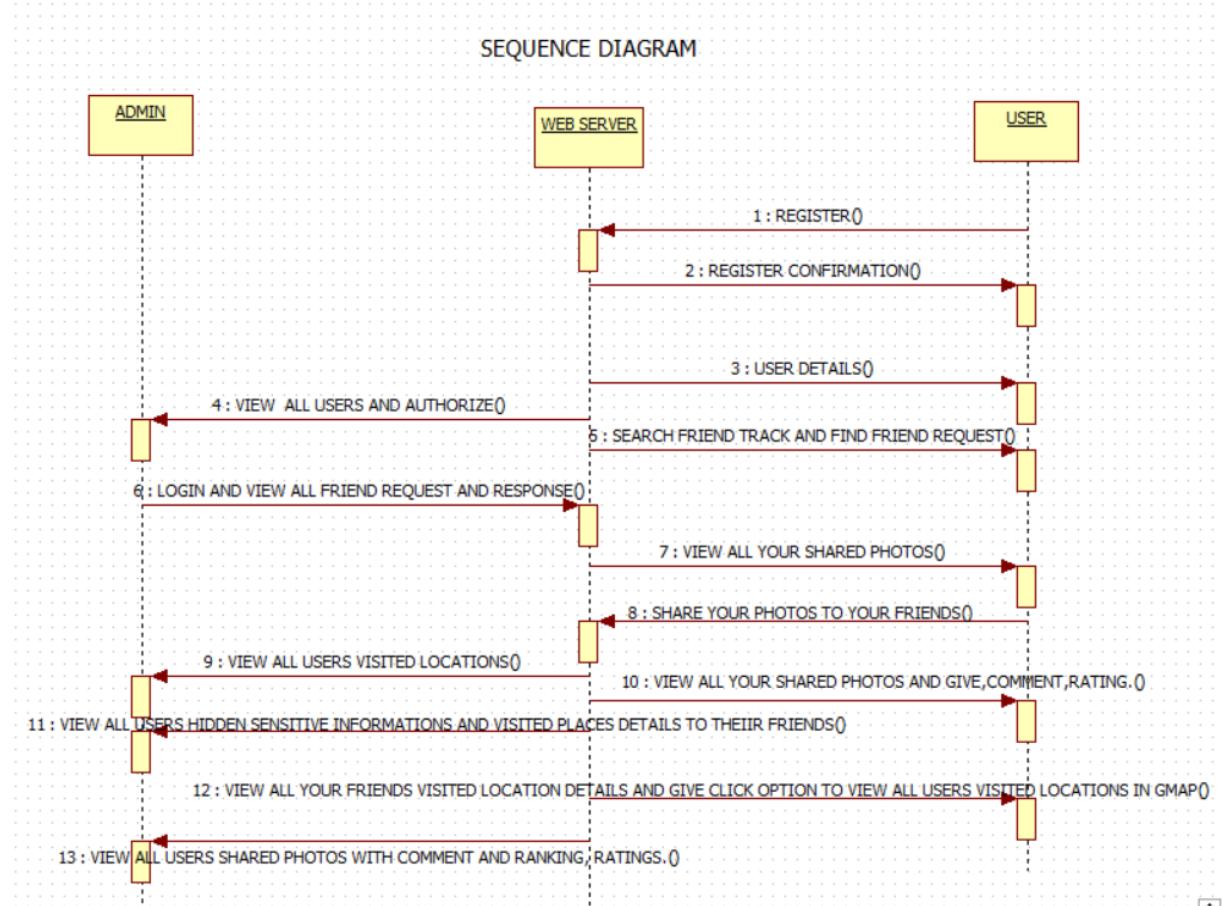


Fig.3. Sequence Diagram.

### 5.2.3 Class Diagram

An object is any person, place, thing, concept, event, screen, or report applicable to your system. Objects both know things (they have attributes) and they do things (they have methods).

A class is a representation of an object and, in many ways; it is simply a template from which objects are created. Classes form the main building blocks of an object-oriented application. Although thousands of students attend the university, you would only model one class, called *Student*, which would represent the entire collection of students.

Classes are typically modeled as rectangles with three sections: the top section for the name of the class, the middle section for the attributes of the class, and the bottom section for the methods of the class. Attributes are the information stored about an object, while methods are the things an object or class do.

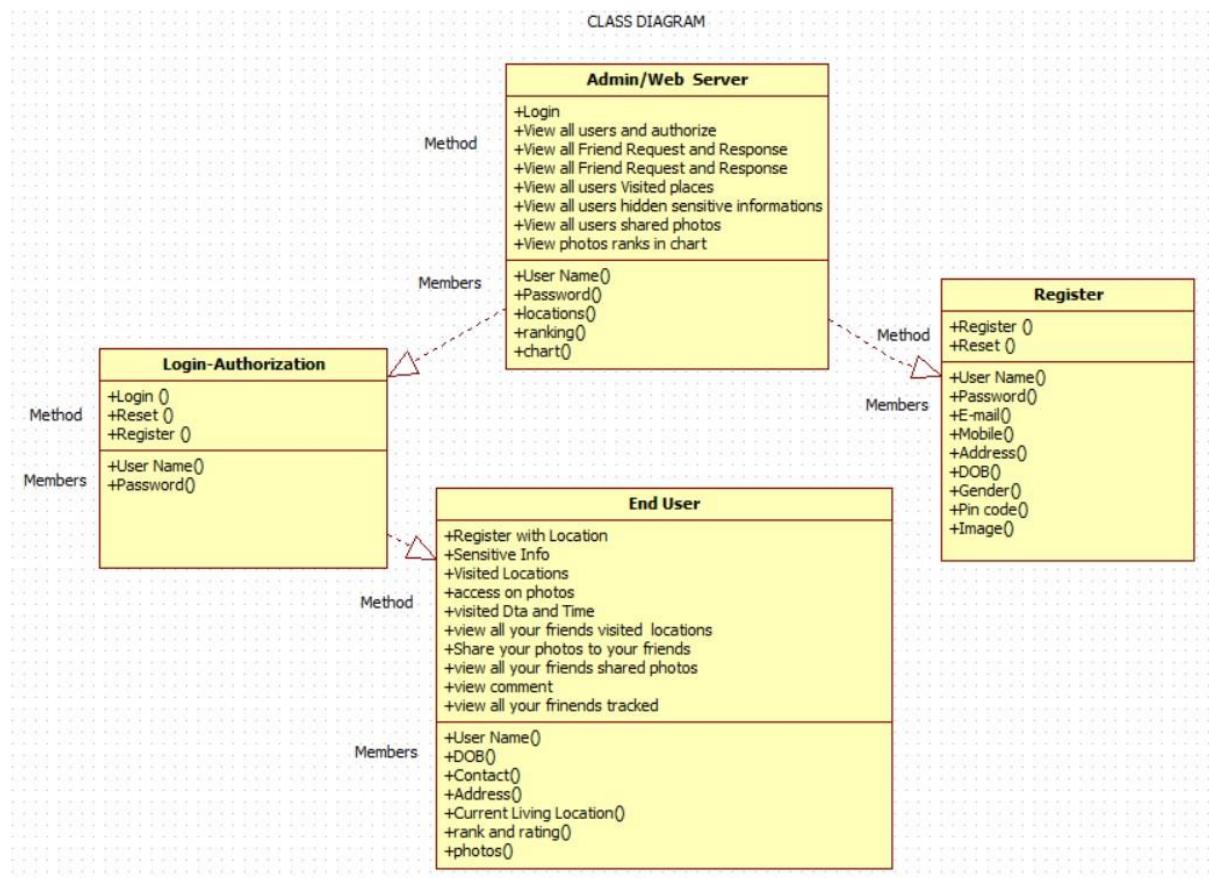


Fig.4. Class Diagram.

#### **5.2.4 Activity Diagrams**

Activity diagrams represent the business and operational work-flows of a system. An Activity diagram is a dynamic diagram that shows the activity and the event that causes the object to be in the particular state.

So, what is the importance of an Activity diagram, as opposed to a State diagram? A State diagram shows the different states an object is in during the lifecycle of its existence in the system, and the transitions in the states of the objects. These transitions depict the activities causing these transitions, shown by arrows.

An Activity diagram talks more about these transitions and activities causing the changes in the object states.

#### 5.2.4.1 Activity Diagram of Admin

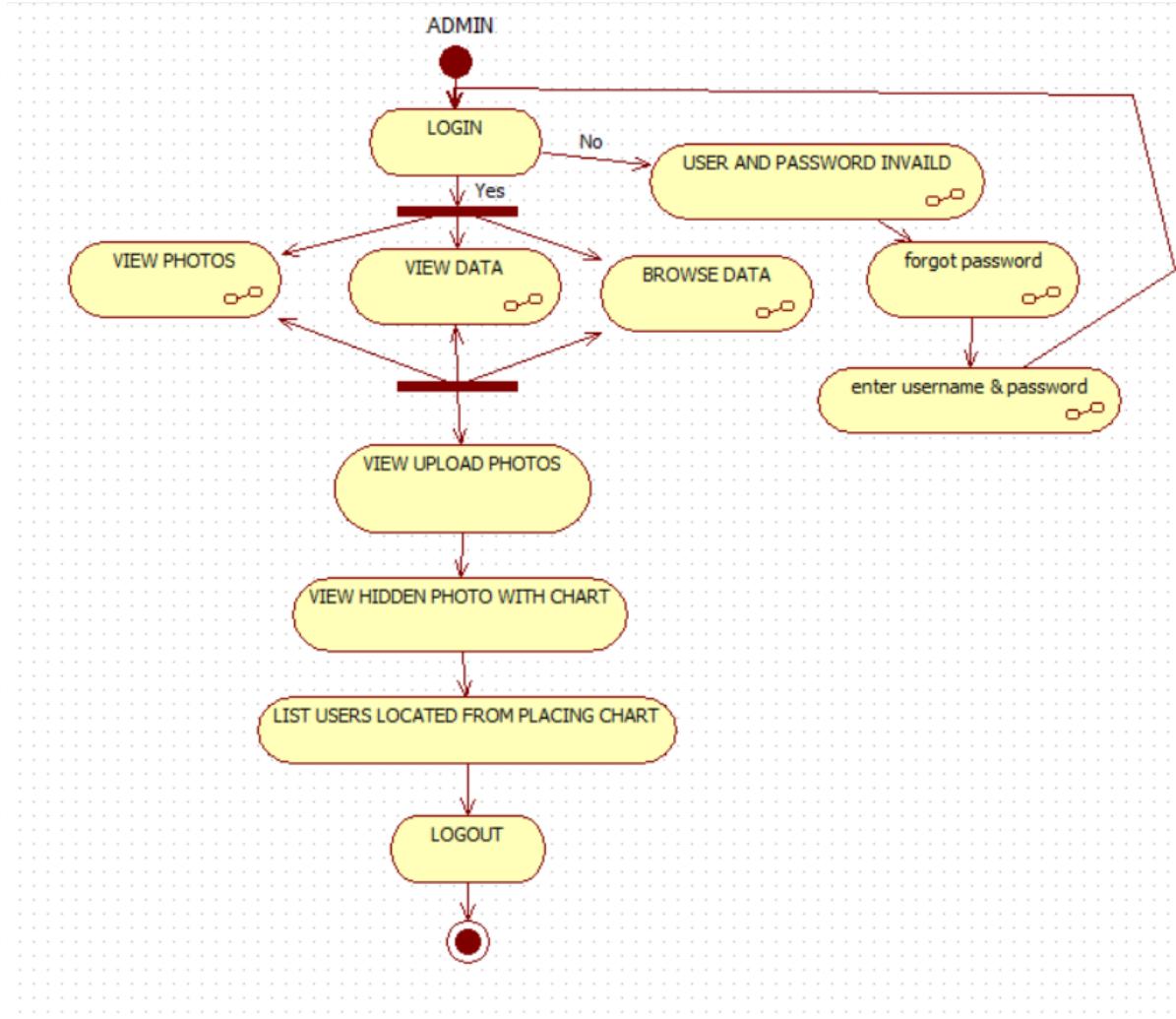


Fig.5 Activity Diagram of Admin.

#### 5.2.4.2 Activity diagram of User

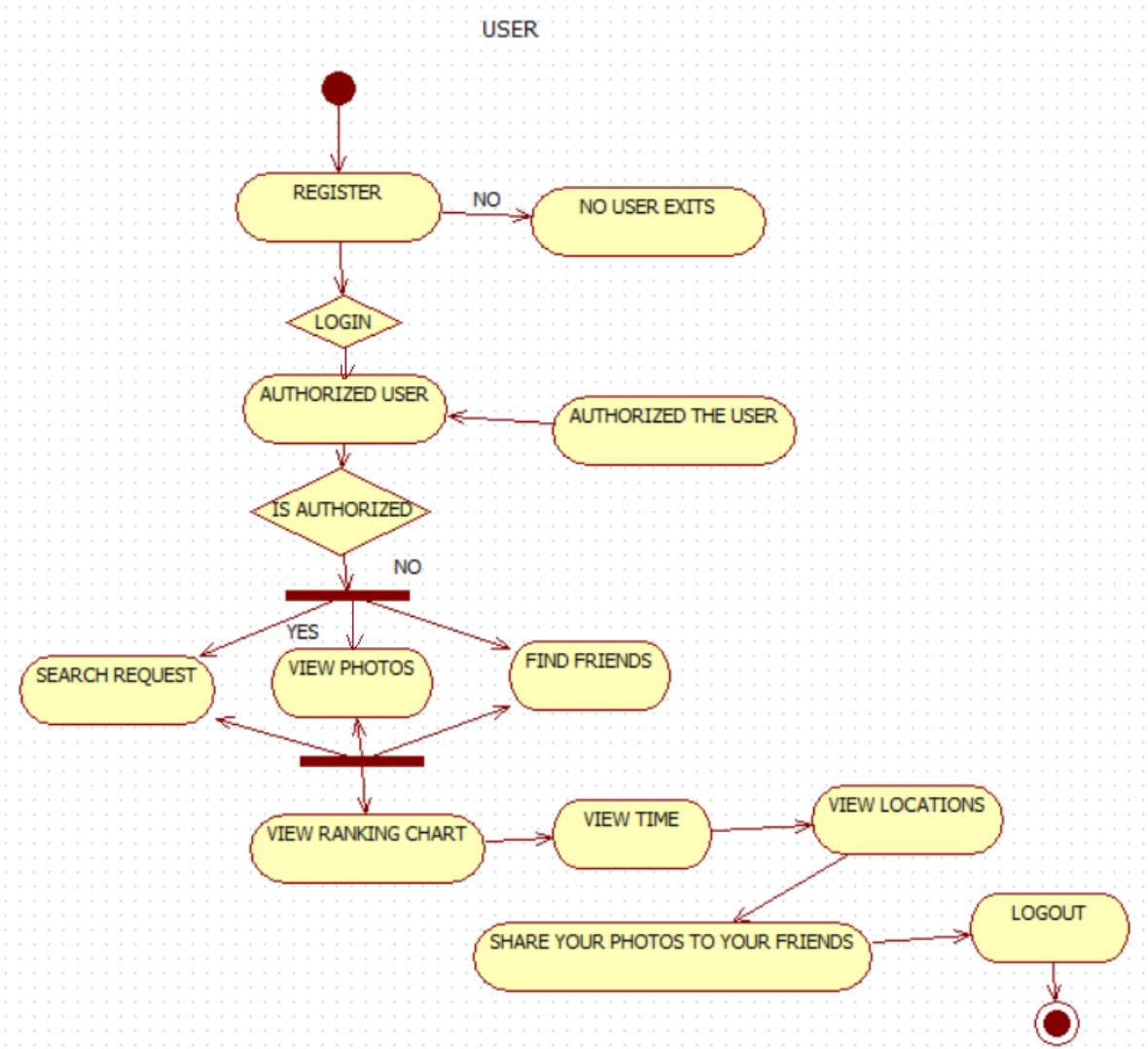


Fig.6. Activity diagram of User.

## **6. SOFTWARE ENVIRONMENT**

### **6.1 Client server overview**

With the varied topic in existence in the fields of computers, Client Server is one, which has generated more heat than light, and also more hype than reality. This technology has acquired a certain critical mass attention with its dedication conferences and magazines. Major computer vendors such as IBM and DEC, have declared that Client Servers is their main future market. A survey of DBMS magazine revealed that 76% of its readers were actively looking at the client server solution. The growth in the client server development tools from \$200 million in 1992 to more than \$1.2 billion in 1996. Client server implementations are complex but the underlying concept is simple and powerful. A client is an application running with local resources but able to request the database and relate the services from separate remote server. The software mediating this client server interaction is often referred to as MIDDLEWARE. The typical client either a PC or a Work Station connected through a network to a more powerful PC, Workstation, Midrange or Main Frames server usually capable of handling request from more than one client. However, with some configuration server may also act as client. A server may need to access other server in order to process the original client request. The key client server idea is that client as user is essentially insulated from the physical location and formats of the data needs for their application. With the proper middleware, a client input from or report can transparently access and manipulate both local database on the client machine and remote databases on one or more servers. An added bonus is the client server opens the door to multi-vendor database access indulging heterogeneous table joins.

### **6.2 What is a Client Server?**

Two prominent systems in existence are client server and file server systems. It is essential to distinguish between client servers and file server systems. Both provide shared network access to data but the comparison ends there! The file server simply provides a remote disk drive that can be accessed by LAN applications on a file by file basis. The client server offers full relational database services such as SQL-Access, Record modifying, Insert, Delete with full relational integrity backup/ restore performance for high volume of transactions, etc. the client

server middleware provides a flexible interface between client and server, who does what, when and to whom.

### **6.3 Why Client Server?**

Client server has evolved to solve a problem that has been around since the earliest days of computing: how best to distribute your computing, data generation and data storage resources in order to obtain efficient, cost effective departmental an enterprise wide data processing. During mainframe era choices were quite limited. A central machine housed both the CPU and DATA (cards, tapes, drums and later disks). Access to these resources was initially confined to batched runs that produced departmental reports at the appropriate intervals. A strong central information service department ruled the corporation. The role of the rest of the corporation limited to requesting new or more frequent reports and to provide hand written forms from which the central data banks were created and updated. The earliest client server solutions therefore could best be characterized as “SLAVE-MASTER”.

Time-sharing changed the picture. Remote terminal could view and even change the central data, subject to access permissions. And, as the central data banks evolved in to sophisticated relational database with non-programmer query languages, online users could formulate adhoc queries and produce local reports without adding to the MIS applications software backlog. However remote access was through dumb terminals, and the client server remained subordinate to the Slave\Master.

### **6.4 Front-End/User Interface Design**

The entire user interface is planned to be developed in browser specific environment with a touch of Intranet-Based Architecture for achieving the Distributed Concept. The browser specific components are designed by using the HTML standards, and the dynamism of the designed by concentrating on the constructs of the Java Server Pages.

### **6.5 Communication/Database Connectivity Tier**

The Communication architecture is designed by concentrating on the Standards of Servlets and Enterprise Java Beans. The database connectivity is established by using the Java Data Base Connectivity. The standards of three-tier architecture are given major concentration to keep the standards of higher cohesion and limited coupling for effectiveness of the operations.

## **6.6 Features of the Language Used**

In our project, we have chosen *Java* language for developing the code.

### **About Java:**

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

- Java is a programmer’s language.
- Java is cohesive and consistent.
- Except for those constraints imposed by the Internet environment, Java gives the programmer, full control.

Finally, Java is to Internet programming where C was to system programming.

## **6.7 Importance of Java to the Internet**

Java has had a profound effect on the Internet. This is because; Java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the Server and the Personal computer. They are: Passive information and Dynamic active programs. The Dynamic, Self-executing programs cause serious problems in the areas of Security and probability. But, Java addresses those concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

## **6.8 Java can be used to create two types of programs**

### **Applications and applets**

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

## **Features of Java**

### **Security**

Every time you download a “normal” program; you are risking a viral infection. Prior to Java, most users did not download executable programs frequently, and those who did scanned them for viruses prior to execution. Most users still worried about the possibility of infecting their systems with a virus. In addition, another type of malicious program exists that must be guarded against. This type of program can gather private information, such as credit card numbers, bank account balances, and passwords. Java answers both these concerns by providing a “firewall” between a network application and your computer.

When you use a Java-compatible Web browser, you can safely download Java applets without fear of virus infection or malicious intent.

### **Portability**

For programs to be dynamically downloaded to all the various types of platforms connected to the Internet, some means of generating portable executable code is needed. As you will see, the same mechanism that helps ensure security also helps create portability. Indeed, Java’s solution to these two problems is both elegant and efficient.

### **The Byte code**

The key that allows the Java to solve the security and portability problems is that the output of Java compiler is Byte code. Byte code is a highly optimized set of instructions designed to be executed by the Java run-time system, which is called the Java Virtual Machine (JVM). That is, in its standard form, the JVM is an interpreter for byte code.

Translating a Java program into byte code helps makes it much easier to run a program in a wide variety of environments. The reason is, once the run-time package exists for a given system, any Java program can run on it.

Although Java was designed for interpretation, there is technically nothing about Java that prevents on-the-fly compilation of byte code into native code. Sun has just completed its Just In Time (JIT) compiler for byte code. When the JIT compiler is a part of JVM, it compiles byte code into executable code in real time, on a piece-by-piece, demand basis. It is not possible to compile an entire Java program into executable code all at once, because Java performs various

run-time checks that can be done only at run time. The JIT compiles code, as it is needed, during execution.

## Java Virtual Machine (JVM)

Beyond the language, there is the Java virtual machine. The Java virtual machine is an important element of the Java technology. The virtual machine can be embedded within a web browser or an operating system. Once a piece of Java code is loaded onto a machine, it is verified. As part of the loading process, a class loader is invoked and does byte code verification makes sure that the code that's has been generated by the compiler will not corrupt the machine that it's loaded on. Byte code verification takes place at the end of the compilation process to make sure that is all accurate and correct. So byte code verification is integral to the compiling and executing of Java code.



Java programming uses to produce byte codes and executes them. The first box indicates that the Java source code is located in a. Java file that is processed with a Java compiler called javac. The Java compiler produces a file called a. class file, which contains the byte code. The Class file is then loaded across the network or loaded locally on your machine into the execution environment is the Java virtual machine, which interprets and executes the byte code.

## Java Architecture

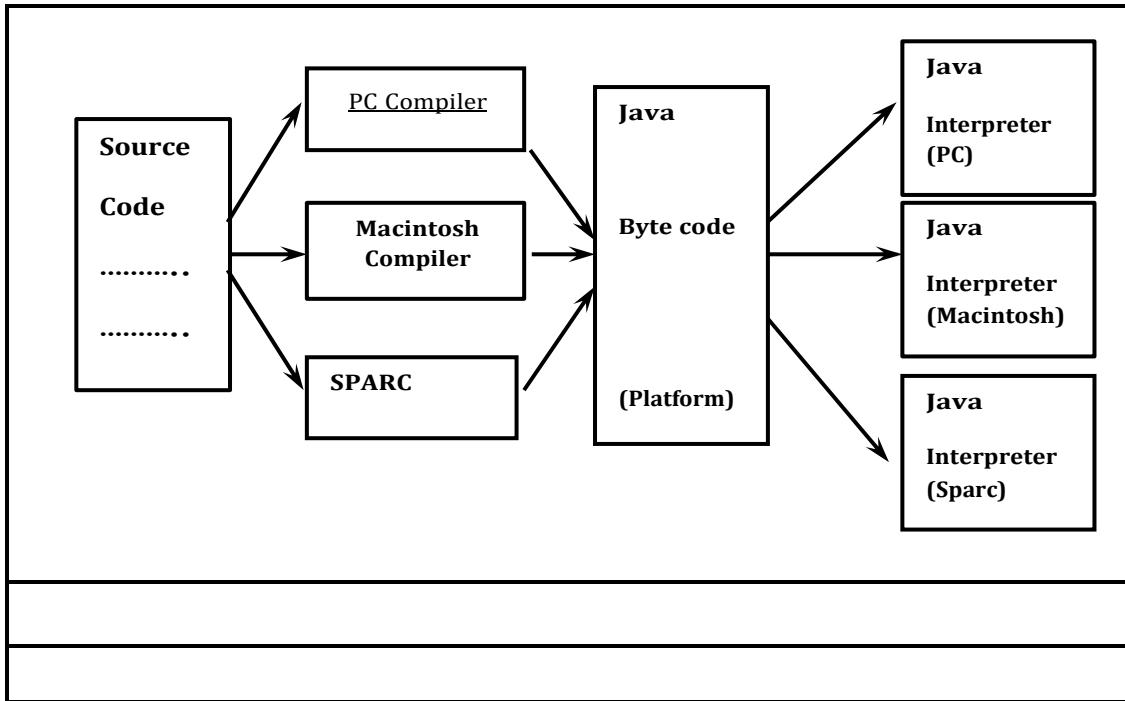
Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed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

## Compilation of code

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

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

## Compiling and interpreting Java Source Code



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

## **Simple**

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

## **Object-Oriented**

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank slate. One outcome of this was a clean usable,

pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

## **Robust**

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

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

## **JAVASCRIPT**

JavaScript is a script-based programming language that was developed by Netscape Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java. JavaScript supports the development of both client and server components of Web-based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write Web server programs that can process information submitted by a Web browser and then updates the browser's display accordingly.

Even though JavaScript supports both client and server Web programming, we prefer JavaScript at Client side programming since most of the browsers supports it. JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags.

<SCRIPT>..</SCRIPT>

<SCRIPT LANGUAGE = “JavaScript”>

JavaScript statements

</SCRIPT>

Here are a few things we can do with JavaScript:

- Validate the contents of a form and make calculations.
- Add scrolling or changing messages to the Browser’s status line.

- Animate images or rotate images that change when we move the mouse over them.
- Detect the browser in use and display different content for different browsers.
- Detect installed plug-ins and notify the user if a plug-in is required.

We can do much more with JavaScript, including creating entire application.

## **JAVASCRIPT VS JAVA**

JavaScript and Java are entirely different languages. A few of the most glaring differences are:

- Java applets are generally displayed in a box within the web document; JavaScript can affect any part of the Web document itself.
- While JavaScript is best suited to simple applications and adding interactive features to Web pages; Java can be used for incredibly complex applications.

There are many other differences but the important thing to remember is that JavaScript and Java are separate languages. They are both useful for different things; in fact they can be used together to combine their advantages.

### **ADVANTAGES:**

JavaScript can be used for Sever-side and Client-side scripting.

- It is more flexible than VBScript.
- JavaScript is the default scripting languages at Client-side since all the browsers supports it.

## **Hyper Text Markup Language**

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

HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each

delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

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

HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

### **Basic HTML Tags:**

<b>&lt;!-- --&gt;</b>	Specifies comments
<b>&lt;A&gt;.....&lt;/A&gt;</b>	Creates hypertext links
<b>&lt;B&gt;.....&lt;/B&gt;</b>	Formats text as bold
<b>&lt;BODY&gt;...&lt;/BODY&gt;</b>	Contains all tags and text in the HTML document
<b>&lt;CENTER&gt;...&lt;/CENTER&gt;</b>	Creates text
<b>&lt;DD&gt;...&lt;/DD&gt;</b>	Definition of a term
<b>&lt;DL&gt;...&lt;/DL&gt;</b>	Creates definition list
<b>&lt;BIG&gt;.....&lt;/BIG&gt;</b>	Formats text in large font.
<b>&lt;FONT&gt;...&lt;/FONT&gt;</b>	Formats text with a particular font
<b>&lt;FORM&gt;...&lt;/FORM&gt;</b>	Encloses a fill-out form
<b>&lt;FRAME&gt;...&lt;/FRAME&gt;</b>	Defines a particular frame in a set of frames
<b>&lt;H#&gt;...&lt;/H#&gt;</b>	Creates headings of different levels
<b>&lt;HR&gt;...&lt;/HR&gt;</b>	Creates a horizontal rule
<b>&lt;HTML&gt;...&lt;/HTML&gt;</b>	Contains all other HTML tags
<b>&lt;META&gt;...&lt;/META&gt;</b>	Provides meta-information about a document

<b>&lt;SCRIPT&gt;...&lt;/SCRIPT&gt;</b>	Contains client-side or server-side script
<b>&lt;TABLE&gt;...&lt;/TABLE&gt;</b>	Creates a table
<b>&lt;TD&gt;...&lt;/TD&gt;</b>	Indicates table data in a table
<b>&lt;TR&gt;...&lt;/TR&gt;</b>	Designates a table row
<b>&lt;TH&gt;...&lt;/TH&gt;</b>	Creates a heading in a table

## ADVANTAGES

A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.

- HTML is platform independent.
- HTML tags are not case-sensitive.

## Java Database Connectivity

### What Is JDBC?

JDBC is a Java API for executing SQL statements. (As a point of interest, JDBC is a trademarked name and is not an acronym; nevertheless, JDBC is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the Java programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API.

Using JDBC, it is easy to send SQL statements to virtually any relational database. One can write a single program using the JDBC API, and the program will be able to send SQL statements to the appropriate database. The combinations of Java and JDBC lets a programmer write it once and run it anywhere.

### What Does JDBC Do?

Simply put, JDBC makes it possible to do three things:

- Establish a connection with a database
- Send SQL statements

- Process the results.

## JDBC versus ODBC and other APIs:

At this point, Microsoft's ODBC (Open Database Connectivity) API is probably the most widely used programming interface for accessing relational databases. It offers the ability to connect to almost all databases on almost all platforms.

So why not just use ODBC from Java? The answer is that you can use ODBC from Java, but this is best done with the help of JDBC in the form of the JDBC-ODBC Bridge, which we will cover shortly. The question now becomes "Why do you need JDBC?" There are several answers to this question:

ODBC is not appropriate for direct use from Java because it uses a C interface. Calls from Java to native C code have a number of drawbacks in the security, implementation, robustness, and automatic portability of applications.

A literal translation of the ODBC C API into a Java API would not be desirable. For example, Java has no pointers, and ODBC makes copious use of them, including the notoriously error-prone generic pointer "void \*". You can think of JDBC as ODBC translated into an object-oriented interface that is natural for Java programmers.

ODBC is hard to learn. It mixes simple and advanced features together, and it has complex options even for simple queries. JDBC, on the other hand, was designed to keep simple things simple while allowing more advanced capabilities where required.

A Java API like JDBC is needed in order to enable a "pure Java" solution. When ODBC is used, the ODBC driver manager and drivers must be manually installed on every client machine. When the JDBC driver is written completely in Java, however, JDBC code is automatically installable, portable, and secure on all Java platforms from network computers to mainframes.

### Two-tier and Three-tier Models

The JDBC API supports both two-tier and three-tier models for database access. In the two-tier model, a Java applet or application talks directly to the database.

This requires a JDBC driver that can communicate with the particular database management system being accessed. A user's SQL statements are delivered to the database, and the results of those statements are sent back to the user. The database may be located on another machine to which the user is connected via a network. This is referred to as a client/server configuration, with the user's machine as the client, and the machine housing the database as the server. The network can be an Intranet, which, for example, connects employees within a corporation, or it can be the Internet.

In the three-tier model, commands are sent to a "middle tier" of services, which then send SQL statements to the database. The database processes the SQL statements and sends the results

back to the middle tier, which then sends them to the user. MIS directors find the three-tier model very attractive because the middle tier makes it possible to maintain control over access and the kinds of updates that can be made to corporate data. Another advantage is that when there is a middle tier, the user can employ an easy-to-use higher-level API which is translated by the middle tier into the appropriate low-level calls. Finally, in many cases the three-tier architecture can provide performance advantages.

Until now the middle tier has typically been written in languages such as C or C++, which offer fast performance. However, with the introduction of optimizing compilers that translate Java byte code into efficient machine-specific code, it is becoming practical to implement the middle tier in Java. This is a big plus, making it possible to take advantage of Java's robustness, multithreading, and security features. JDBC is important to allow database access from a Java middle tier.

## **JDBC Driver Types**

The JDBC drivers that we are aware of at this time fit into one of four categories:

- JDBC-ODBC bridge plus ODBC driver
- Native-API partly-Java driver
- JDBC-Net pure Java driver
- Native-protocol pure Java driver

## **JDBC-ODBC Bridge**

If possible, use a Pure Java JDBC driver instead of the Bridge and an ODBC driver. This completely eliminates the client configuration required by ODBC. It also eliminates the potential that the Java VM could be corrupted by an error in the native code brought in by the Bridge (that is, the Bridge native library, the ODBC driver manager library, the ODBC driver library, and the database client library).

## **What Is the JDBC- ODBC Bridge?**

The JDBC-ODBC Bridge is a JDBC driver, which implements JDBC operations by translating them into ODBC operations. To ODBC it appears as a normal application program. The Bridge implements JDBC for any database for which an ODBC driver is available. The Bridge is implemented as the

`sun.jdbc.odbc` Java package and contains a native library used to access ODBC. The Bridge is a joint development of Intervolve and Java Soft.

## **Java Server Pages (JSP)**

Java server Pages is a simple, yet powerful technology for creating and maintaining dynamic-content web pages. Based on the Java programming language, Java Server Pages offers proven portability, open standards, and a mature re-usable component model .The Java Server Pages architecture enables the separation of content generation from content presentation. This separation not eases maintenance headaches, it also allows web team members to focus on their areas of expertise. Now, web page designer can concentrate on layout, and web application designers on programming, with minimal concern about impacting each other's work.

## **Features of JSP**

### **Portability**

Java Server Pages files can be run on any web server or web-enabled application server that provides support for them. Dubbed the JSP engine, this support involves recognition, translation, and management of the Java Server Page lifecycle and its interaction components.

### **Components**

It was mentioned earlier that the Java Server Pages architecture can include reusable Java components. The architecture also allows for the embedding of a scripting language directly into the Java Server Pages file. The components current supported include Java Beans, and Servlets.

### **Processing**

A Java Server Pages file is essentially an HTML document with JSP scripting or tags. The Java Server Pages file has a JSP extension to the server as a Java Server Pages file. Before the page is served, the Java Server Pages syntax is parsed and processed into a Servlet on the server side. The Servlet that is generated outputs real content in straight HTML for responding to the client.

### **Access Models**

A Java Server Pages file may be accessed in at least two different ways. A client's request comes directly into a Java Server Page. In this scenario, suppose the page accesses reusable

Java Bean components that perform particular well-defined computations like accessing a database. The result of the Beans computations, called result sets is stored within the Bean as properties. The page uses such Beans to generate dynamic content and present it back to the client.

In both of the above cases, the page could also contain any valid Java code. Java Server Pages architecture encourages separation of content from presentation.

## **Steps in the execution of a JSP Application**

1. The client sends a request to the web server for a JSP file by giving the name of the JSP file within the form tag of a HTML page.
2. This request is transferred to the JavaWebServer. At the server side JavaWebServer receives the request and if it is a request for a jsp file server gives this request to the JSP engine.
3. JSP engine is program which can understand the tags of the jsp and then it converts those tags into a Servlet program and it is stored at the server side. This Servlet is loaded in the memory and then it is executed and the result is given back to the JavaWebServer and then it is transferred back to the result is given back to the JavaWebServer and then it is transferred back to the client.

## **JDBC connectivity**

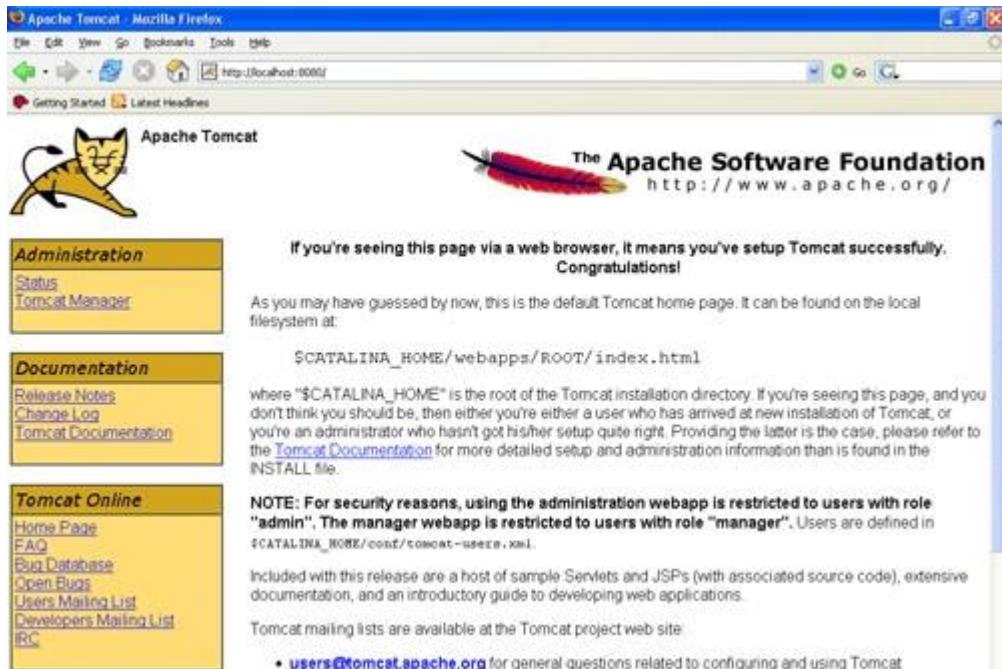
The JDBC provides database-independent connectivity between the J2EE platform and a wide range of tabular data sources. JDBC technology allows an Application Component Provider to:

- Perform connection and authentication to a database server
- Manager transactions
- Move SQL statements to a database engine for preprocessing and execution
- Execute stored procedures
- Inspect and modify the results from Select statements.

## **Tomcat 6.0 web server**

Tomcat is an open source web server developed by Apache Group. Apache Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and

Java Server Pages technologies. The Java Servlet and Java Server Pages specifications are developed by Sun under the Java Community Process. Web Servers like Apache Tomcat support only web components while an application server supports web components as well as business components (BEAs WebLogic, is one of the popular application server). To develop a web application with jsp/servlet install any web server like JRun, Tomcat etc to run your application.



## 7. CODING AND IMPLEMENTATION

### Admin

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Admin</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="css/style.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/droid_sans_400-droid_sans_700.font.js"></script>
<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>
<script type="text/javascript" src="js/script.js"></script>
<script type="text/javascript" src="js/coin-slider.min.js"></script>
<style type="text/css">
<!--
.style1 { font-size: 36px }
-->
</style>
</head>
<body>
<div class="main">
<div class="header">
<div class="header_resize">
<div class="logo">
<h1><a href="index.html"><span class="style1">Follow But No Track-Privacy Preserved
Profile Publishing in Cyber Physical Social Systems</span></a></h1>
</div>
<div class="clr"></div>
<div class="menu_nav">
<ul>
<li><a href="index.html"><span>Home Page</span></a></li>
<li class="active"><a href="A_Login.jsp"><span>Admin</span></a></li>
<li><a href="U_Login.jsp"><span>user</span></a></li>
</ul>
</div>
<div class="clr"></div>
<div class="slider">
<div id="coin-slider"><a href="#"><span><big>Follow But No Track-Privacy Preserved Profile Publishing in Cyber
Physical Social Systems</big></span></a><a href="#"><span><big>Follow But No Track-Privacy Preserved
Profile Publishing in Cyber Physical Social Systems</big></span></a><a
href="#">
```

```

/><span><big>Follow But No Track-Privacy Preserved Profile Publishing in Cyber Physical
Social Systems</big></span></a></div>
<div class="clr"></div>
</div>
<div class="clr"></div>
</div>
</div>
<div class="content">
<div class="content_resize">
<div class="mainbar">
<p>&nbsp;</p>
<p>&nbsp;</p>
</div>
<div class="sidebar">
<div class="searchform">
<form id="formsearch" name="formsearch" method="post" action="#">
<span>
<input name="editbox_search" class="editbox_search" id="editbox_search" maxlength="80"
value="Search our ste:" type="text" />
</span>
<input name="button_search" src="images/search.gif" class="button_search" type="image" />
</form>
</div>
<div class="gadget">
<h2 class="star"><span>Sidebar</span> Menu</h2>
<div class="clr"></div>
<ul class="sb_menu">
<li><a href="A_Main.jsp">Admin Home</a></li>
<li><a href="A_Login.jsp">Logout</a></li>
</ul>
</div>
</div>
<div class="clr"></div>
</div>
</div>
<div class="fbg">
<div class="fbg_resize">
<div class="col c1">
<h2><span>Image</span> Gallery</h2>
<a href="#"><imgsrc="images/gal1.jpg" width="75" height="75" alt="" class="gal"
/></a><a href="#"><imgsrc="images/gal2.jpg" width="75" height="75" alt="" class="gal"
/></a><a href="#"><imgsrc="images/gal3.jpg" width="75" height="75" alt="" class="gal"
/></a><a href="#"><imgsrc="images/gal4.jpg" width="75" height="75" alt="" class="gal"
/></a><a href="#"><imgsrc="images/gal5.jpg" width="75" height="75" alt="" class="gal"
/></a><a href="#"><imgsrc="images/gal6.jpg" width="75" height="75" alt="" class="gal"
/></a></div>
<div class="clr"></div>
</div>
</div>

```

```

<div class="footer">
<div class="footer_resize">
<div style="clear:both;"></div>
</div>
</div>
</div>
<div align=center></div>
</body>
</html>

```

## User

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>User</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="css/style.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/droid_sans_400-droid_sans_700.font.js"></script>
<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>
<script type="text/javascript" src="js/script.js"></script>
<script type="text/javascript" src="js/coin-slider.min.js"></script>
<style type="text/css">
<!--
.style1 {font-size: 36px}
-->
</style>
</head>
<body>
<div class="main">
<div class="header">
<div class="header_resize">
<div class="logo">
<h1><a href="index.html"><span class="style1">Follow But No Track-Privacy Preserved
Profile Publishing in Cyber Physical Social Systems</span></a></h1>
</div>
<div class="clr"></div>
<div class="menu_nav">
<ul>
<li><a href="index.html"><span>Home Page</span></a></li>
<li><a href="A_Login.jsp"><span>Admin</span></a></li>
<li class="active"><a href="U_Login.jsp"><span>user</span></a></li>
</ul>
</div>
<div class="clr"></div>
<div class="slider">

```

```

<div id="coin-slider"><a href="#"><imgsrc="images/slide1.jpg" width="960" height="360" alt="" /><span><big>Follow But No Track-Privacy Preserved Profile Publishing in Cyber Physical Social Systems</big></span></a><a href="#"><imgsrc="images/slide2.jpg" width="960" height="360" alt="" /><span><big>Follow But No Track-Privacy Preserved Profile Publishing in Cyber Physical Social Systems</big></span></a><a href="#"><imgsrc="images/slide3.jpg" width="960" height="360" alt="" /><span><big>Follow But No Track-Privacy Preserved Profile Publishing in Cyber Physical Social Systems</big></span></a></div>
<div class="clr"></div>
</div>
<div class="clr"></div>
</div>
</div>
<div class="content">
<div class="content_resize">
<div class="mainbar">
<p>&nbsp;</p>
<p>&nbsp;</p>
</div>
<div class="sidebar">
<div class="searchform">
<form id="formsearch" name="formsearch" method="post" action="#">
<span>
<input name="editbox_search" class="editbox_search" id="editbox_search" maxlength="80" value="Search our ste:" type="text" />
</span>
<input name="button_search" src="images/search.gif" class="button_search" type="image" />
</form>
</div>
<div class="gadget">
<h2 class="star"><span>Sidebar</span> Menu</h2>
<div class="clr"></div>
<ul class="sb_menu">
<li><a href="U_Main.jsp">User Home</a></li>
<li><a href="U_Login.jsp">Logout</a></li>
</ul>
</div>
</div>
<div class="clr"></div>
</div>
</div>
<div class="fbg">
<div class="fbg_resize">
<div class="col c1">
<h2><span>Image</span> Gallery</h2>
<a href="#"><imgsrc="images/gal1.jpg" width="75" height="75" alt="" class="gal" /></a><a href="#"><imgsrc="images/gal2.jpg" width="75" height="75" alt="" class="gal" /></a><a href="#"><imgsrc="images/gal3.jpg" width="75" height="75" alt="" class="gal" /></a><a href="#"><imgsrc="images/gal4.jpg" width="75" height="75" alt="" class="gal" /></a>

```

```

/></a><a href="#"><imgsrc="images/gal5.jpg" width="75" height="75" alt="" class="gal"
/></a><a href="#"><imgsrc="images/gal6.jpg" width="75" height="75" alt="" class="gal"
/></a></div>
<div class="clr"></div>
</div>
</div>
<div class="footer">
<div class="footer_resize">
<div style="clear:both;"></div>
</div>
</div>
</div align=center></div>
</body>
</html>

```

## Login

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Admin Login Page</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="css/style.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/droid_sans_400-droid_sans_700.font.js"></script>
<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>
<script type="text/javascript" src="js/script.js"></script>
<script type="text/javascript" src="js/coin-slider.min.js"></script>
<style type="text/css">
<!--
.style1 {font-size: 36px}
.style2 {color: #FF00FF}
-->
</style>
<script language="javascript" type="text/javascript">
function valid()
{
var na3=document.s.adminid.value;
if(na3=="")
{
alert("Please Enter User Name");
document.s.adminid.focus();
return false;
}
else
{

```

```

}

var na4=document.s.pass.value;
if(na4=="")
{

alert("Please Enter Password");
document.s.pass.focus();
return false;
}

}

</script>
</head>
<body>
<div class="main">
<div class="header">
<div class="header_resize">
<div class="logo">
<h1><a href="index.html"><span class="style1">Follow But No Track-Privacy  
Preserved Profile Publishing in Cyber Physical Social Systems</span></a></h1>
</div>
<div class="clr"></div>
<div class="menu_nav">
<ul>
<li><a href="index.html"><span>Home Page</span></a></li>
<li class="active"><a href="A_Login.jsp"><span>Admin</span></a></li>
<li><a href="U_Login.jsp"><span>user</span></a></li>
</ul>
</div>
<div class="clr"></div>
<div class="slider">
<div id="coin-slider"> <a href="#"><span><big>Follow But No Track-Privacy Preserved Profile Publishing in Cyber Physical Social Systems</big></span></a> <a href="#"><span><big>Follow But No Track-Privacy Preserved Profile Publishing in Cyber Physical Social Systems</big></span></a> <a href="#"><span><big>Follow But No Track-Privacy Preserved Profile Publishing in Cyber Physical Social Systems</big></span></a> </div>
<div class="clr"></div>
</div>
<div class="clr"></div>
</div>
</div>
<div class="content">
<div class="content_resize">
<div class="mainbar">
<h2><span class="style2">Welcome to Admin Login</span></h2>

```

```

<p align="center"></p>
<form name="s" action="Authentication.jsp?value=<%="adminlogin"%>">
method="post" onSubmit="return valid()" onsubmit="return valid()" target="_top">
    <table width="660" align="left">
        <tr>
            <td width="162" height="35"><span class="style6 style3 style2">User Name<br/>(required)</span></td>
            <td width="486"><input id="name" name="adminid" class="text" /></td>
        </tr>
        <tr>
            <td height="38"><span class="style6 style3 style2">Password<br/>(required)</span></td>
            <td><input type="password" id="password" name="pass" class="text" /></td>
        </tr>

        <tr><td></td>
        <td>
            <input name="imageField" type="submit" class="style3" id="imageField" value="Login" />
            <input name="Reset" type="reset" class="style3" value="Reset" /></td>
        </tr>

        </p>
        <p>&nbsp;</p>
    </table>
</form>
</div>
<div class="sidebar">
    <div class="searchform">
        <form id="formsearch" name="formsearch" method="post" action="#">
            <span>
                <input name="editbox_search" class="editbox_search" id="editbox_search" maxlength="80" value="Search our site:" type="text" />
            </span>
            <input name="button_search" src="images/search.gif" class="button_search" type="image" />
        </form>
    </div>
    <div class="gadget">
        <h2 class="star"><span>Sidebar</span> Menu</h2>
        <div class="clr"></div>
        <ul class="sb_menu">
            <li><a href="index.html">Home</a></li>
            <li><a href="A_Login.jsp">Admin</a></li>
            <li><a href="U_Login.jsp">User</a></li>
        </ul>
    </div>
</div>
<div class="clr"></div>
</div>

```

```

</div>
<div class="fbg">
  <div class="fbg_resize">
    <div class="col c1">
      <h2><span>Image</span> Gallery</h2>
      <a href="#">
    </a><a href="#">
    </a><a href="#">
    </a><a href="#">
    </a><a href="#">
    </a><a href="#">
  </div>
  <div class="clr"></div>
</div>
<div class="footer">
  <div class="footer_resize">
    <div style="clear:both;"></div>
  </div>
</div>
<div align=center></div>
</body>
</html>

```

## Main

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Admin Main Page</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="css/style.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/droid_sans_400-droid_sans_700.font.js"></script>
<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>
<script type="text/javascript" src="js/script.js"></script>
<script type="text/javascript" src="js/coin-slider.min.js"></script>
<style type="text/css">
<!--
.style1 {font-size: 36px}
.style2 {color: #FF00FF}
.style3 {
  font-weight: bold;
  font-style: italic;
  color: #000000;
}
-->

```

```

</style>
</head>
<body>
<div class="main">
<div class="header">
<div class="header_resize">
<div class="logo">
<h1><a href="index.html"><span class="style1">Follow But No Track-Privacy Preserved Profile Publishing in Cyber Physical Social Systems</span></a></h1>
</div>
<div class="clr"></div>
<div class="menu_nav">
<ul>
<li><a href="index.html"><span>Home Page</span></a></li>
<li class="active"><a href="A_Login.jsp"><span>Admin</span></a></li>
<li><a href="U_Login.jsp"><span>user</span></a></li>
</ul>
</div>
<div class="clr"></div>
<div class="slider">
<div id="coin-slider"> <a href="#"><span><big>Follow But No Track-Privacy Preserved Profile Publishing in Cyber Physical Social Systems</big></span></a> <a href="#"><span><big>Follow But No Track-Privacy Preserved Profile Publishing in Cyber Physical Social Systems</big></span></a> <a href="#"><span><big>Follow But No Track-Privacy Preserved Profile Publishing in Cyber Physical Social Systems</big></span></a> </div>
<div class="clr"></div>
</div>
<div class="clr"></div>
</div>
<div class="content">
<div class="content_resize">
<div class="mainbar">
<h2><span class="style2">Wel-Come Admin</span></h2>
<p></p>
</div>
<div class="sidebar">

<div class="gadget">
<h2 class="star"><span>Admin</span> Menu</h2>
<div class="clr"></div>
<ul class="sb_menu style3">
<li><a href="A_AuthorizeUsers.jsp">View All Users And Authorize</a></li>
<li><a href="A_AllFrdReqResp.jsp">View All Friend Request and Response</a></li>
<li><a href="A_AllUserPlaces.jsp">View All users Visited places details </a></li>

```

```

<li><a href="A_UserHidenInf.jsp">View all users hidden
sensitive informations And Visited Locations </a></li>
    <li><a href="A_AllUserSharePhotos.jsp">View all users shared
photos </a></li>
        <li><a href="A_HiddenPhoto.jsp">View all users hidden photos
With Chart</a></li>
            <li><a href="A_HiddenInfChart.jsp">List of Users Hidden
sensitive Info and Users Hidden Visited Places in Chart</a></li>
                <li><a href="A_NumUserVisitInChart.jsp">List of Users
Visited Same Place in Chart</a></li>
                    <li><a href="A_UsrInSamePlaceChrt.jsp">List of Users
Located Same place in Chart</a></li>
                        <li><a href="A_UserPhotoRnkChrt.jsp">View Photos ranks in
chart</a></li>
                            <li><a href="A_Login.jsp">LogOut</a></li>
                        </ul>
                    </div>
                </div>
                <div class="clr"></div>
            </div>
        </div>
        <div class="fbg">
            <div class="fbg_resize">
                <div class="col c1">
                    <h2><span>Image</span> Gallery</h2>
                    <a href="#"></a> <a href="#"></a> <a href="#"></a> <a href="#"></a> <a href="#"></a> <a href="#"></a> </div>
                    <div class="clr"></div>
                </div>
            </div>
            <div class="footer">
                <div class="footer_resize">
                    <div style="clear:both;"></div>
                </div>
            </div>
        <div align=center></div>
    </body>
</html>

```

## Connect

```

<title>Quantifying Political Leaning</title><% @ page import="java.sql.*"%>
<% @ page import="java.util.*" %>
<%

```

```
Connection connection = null;
try {

    Class.forName("com.mysql.jdbc.Driver");
    connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/fbnt","root","root");
    String sql="";

}
catch(Exception e)
{
    System.out.println(e);
}
%>
```

## **8. SYSTEM TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### **TYPES OF TESTS**

#### **Unit testing**

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. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

#### **Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

#### **Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

### **System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

### **White Box Testing**

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

### **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.

### **Unit Testing**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

## **Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

## **Objectives**

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

## **Features to be tested**

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

## **Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

## **Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

## **Test Results:**

All the test cases mentioned above passed successfully. No defects encountered.

## **TESTING METHODOLOGIES**

The following are the Testing Methodologies:

- **Unit Testing.**
- **Integration Testing.**
- **User Acceptance Testing.**

- **Output Testing.**
- **Validation Testing.**

## **Unit Testing**

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

## **Integration Testing**

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

**The following are the types of Integration Testing**

### **1) Top-Down Integration**

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

### **2. Bottom-up Integration**

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing

required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

- The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
- A driver (i.e.) the control program for testing is written to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure

The bottom up approaches tests each module individually and then each module is integrated with a main module and tested for functionality.

## **OTHER TESTING METHODOLOGIES**

### **User Acceptance Testing**

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

### **Output Testing**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

## **Validation Checking**

Validation checks are performed on the following fields.

### **Text Field**

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes an error message.

### **Numeric Field**

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error message. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

A successful test is one that gives out the defects for the inappropriate data and produces an output revealing the errors in the system.

### **Preparation of Test Data**

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 that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

### **Using Live Test Data**

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

### **Using Artificial Test Data**

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

The package “Virtual Private Network” has satisfied all the requirements specified as per software requirement specification and was accepted.

### **USER TRAINING**

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

### **MAINTAINENCE**

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user's requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in

future. The coding and designing is simple and easy to understand which will make maintenance easier.

## **TESTING STRATEGY**

A strategy for system testing integrates system test cases and design techniques into a well-planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation. A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

## **SYSTEM TESTING**

Software once validated must be combined with other system elements (e.g. Hardware, people, and database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

## **UNIT TESTING**

In unit testing different modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. Using the detailed design description as a guide, important Conrail paths are tested to uncover errors within the boundary of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module.

In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future projects can either use or interact with this. The future holds a lot to offer to the development and refinement of this project.

## 9. Output Screens

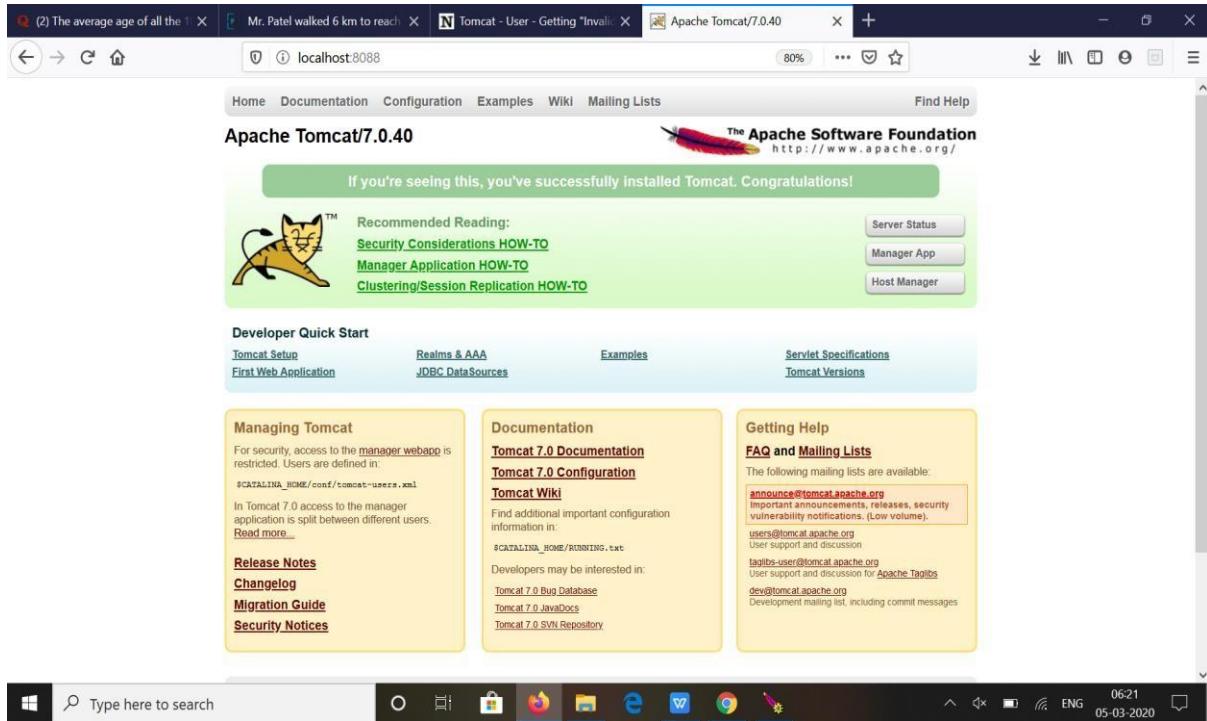
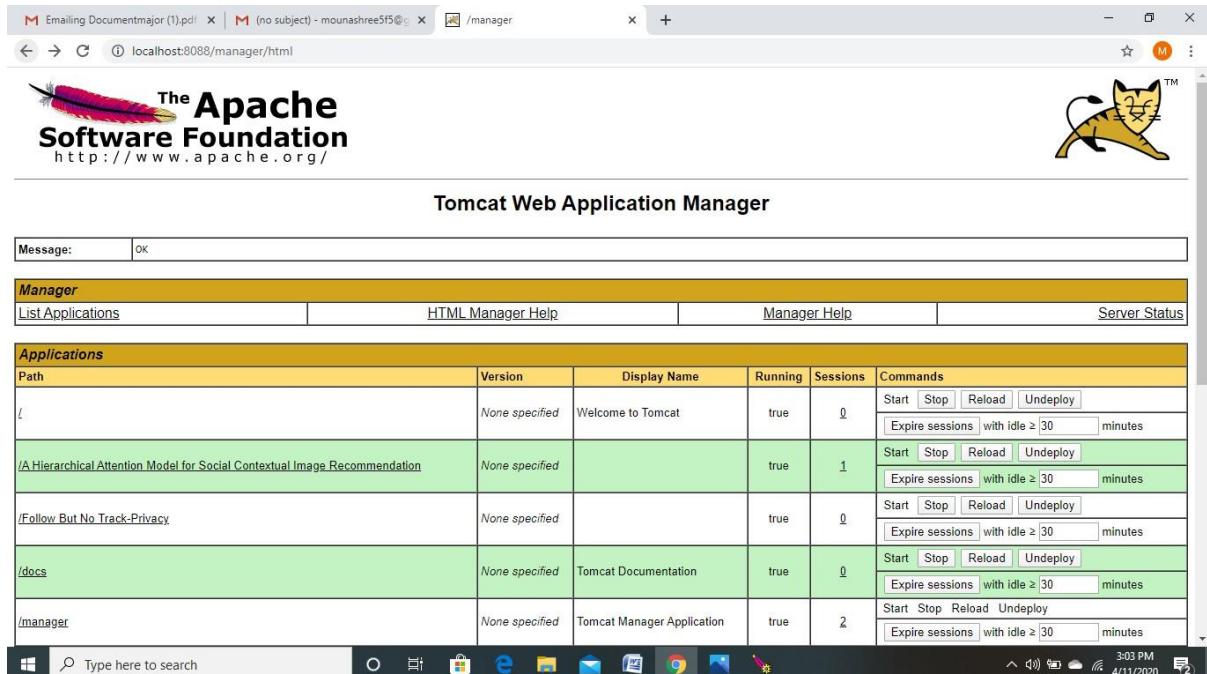


Fig. 1. Apache Tomcat Server.



Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes
/AHierarchical Attention Model for Social Contextual Image Recommendation	None specified		true	1	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes
/Follow But No Track-Privacy	None specified		true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	2	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes

Fig. 2. Tomcat Web Application Manager.

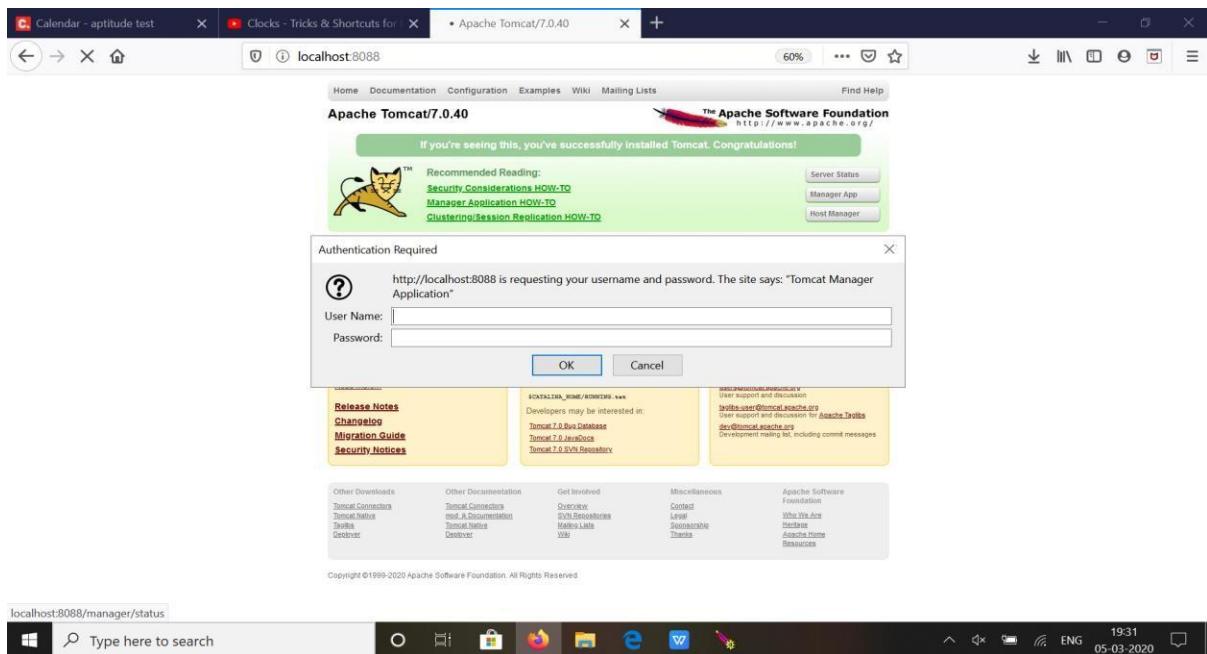


Fig. 3. Apache Tomcat Login Page.



Fig. 4. Home Page

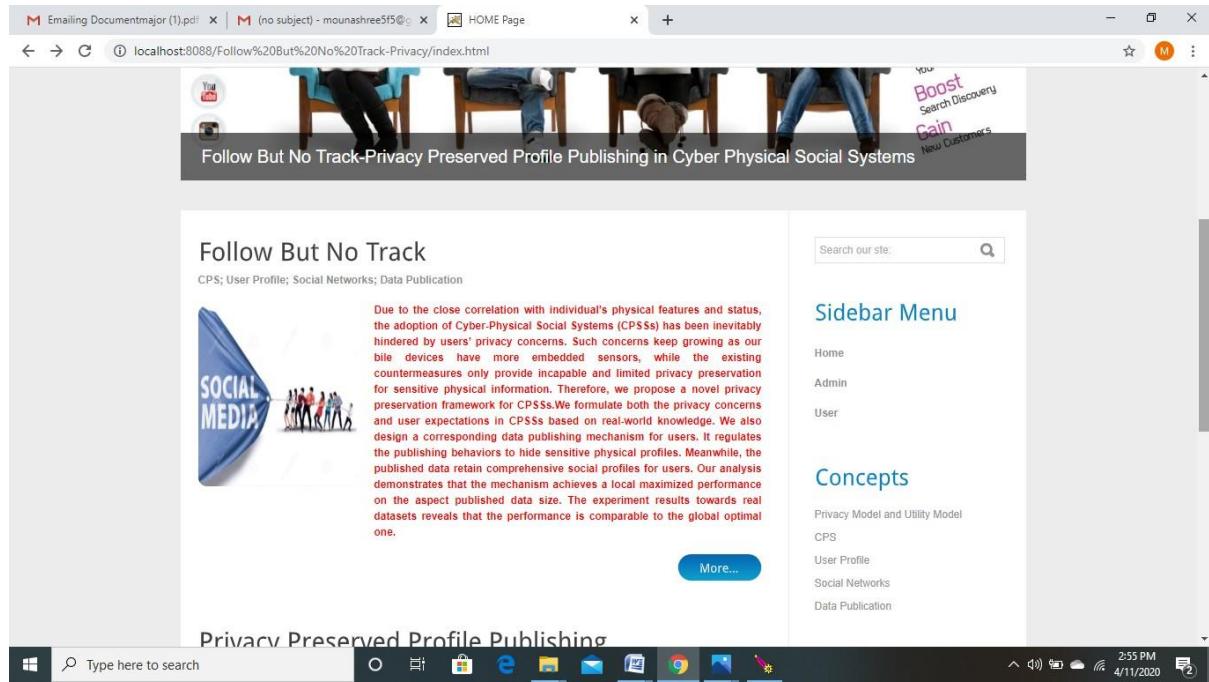


Fig. 5. Home Page.

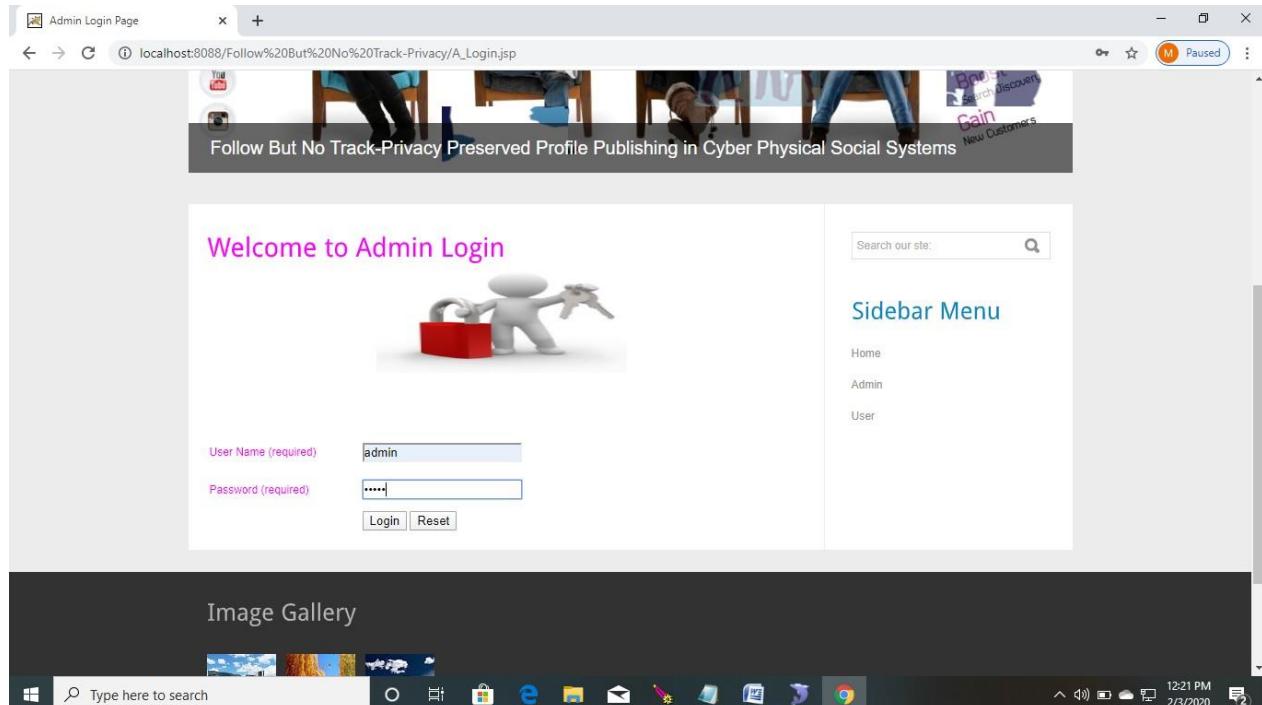


Fig. 6. Admin Login Page.

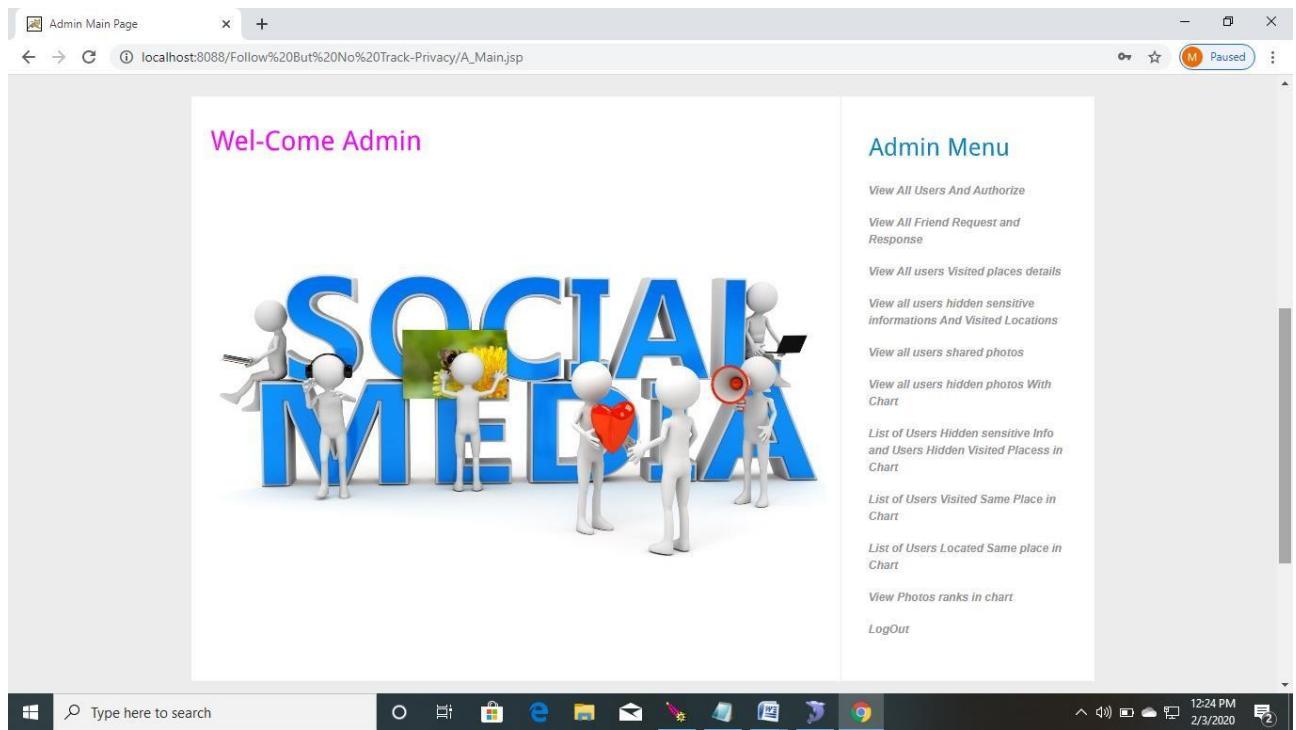


Fig. 7. Admin Home Page.

14		pravalika	fravalikagone@gmail.com	9876543210	Hyderabad	MAQ	Siva	Authorized	<a href="#">View Map</a>
15		Sindhuja	sindhu123@gmail.com	9879879876	Hyderabad	Infosys	Vicky	Authorized	<a href="#">View Map</a>
16		Priyanka	friyankabaru@gmail.com	9898981120	Hyderabad	Microsoft	Nick	Authorized	<a href="#">View Map</a>
17		Kavya	sd@gmail.com	9878685840	Hyderabad	Mind Tree	Single	Authorized	<a href="#">View Map</a>
18		Neha azad	sd@gmail.com	7123654980	Hyderabad	Infosys	Mohith	Authorized	<a href="#">View Map</a>

Fig. 8. View all users place details.

All Request and Response Details...

Username	Request Sent To	Status	Date & Time
rakesh	ramesh	Accepted	10/10/2017 10:51:41
rakesh	omkar	Accepted	10/10/2017 10:51:46
ramesh	omkar	Accepted	10/10/2017 12:25:13
mahesh	omkar	waiting	10/10/2017 12:27:33
Harish	rakesh	Accepted	11/10/2017 12:51:40
Manjunath	rakesh	Accepted	11/10/2017 16:15:36
Manjunath	Harish	Accepted	11/10/2017 16:15:49
sd	Manjunath	waiting	03/02/2020 11:36:36
Fravalika	Friyanka	Accepted	04/02/2020 11:36:45
Fravalika	Rephareddy kavya	waiting	04/02/2020 11:37:01
friyanka	Azamneha	Accepted	04/02/2020 11:39:47
friyanka	Sindhu	Accepted	04/02/2020 11:39:56
sindhu	Fravalika	Accepted	04/02/2020 11:53:12
sindhu	Rephareddy kavya	waiting	04/02/2020 11:53:32
sindhu	Azamneha	Accepted	04/02/2020 11:53:55
Fravalika	Azamneha	waiting	11/03/2020 14:35:27

Back

Fig. 9. All request and response details.

Welcome to User Login

User Name (required):

Password (required):

New User [Click Here To Register](#)

Sidebar Menu

- Home
- Admin
- User

Fig. 10. User Login Page.

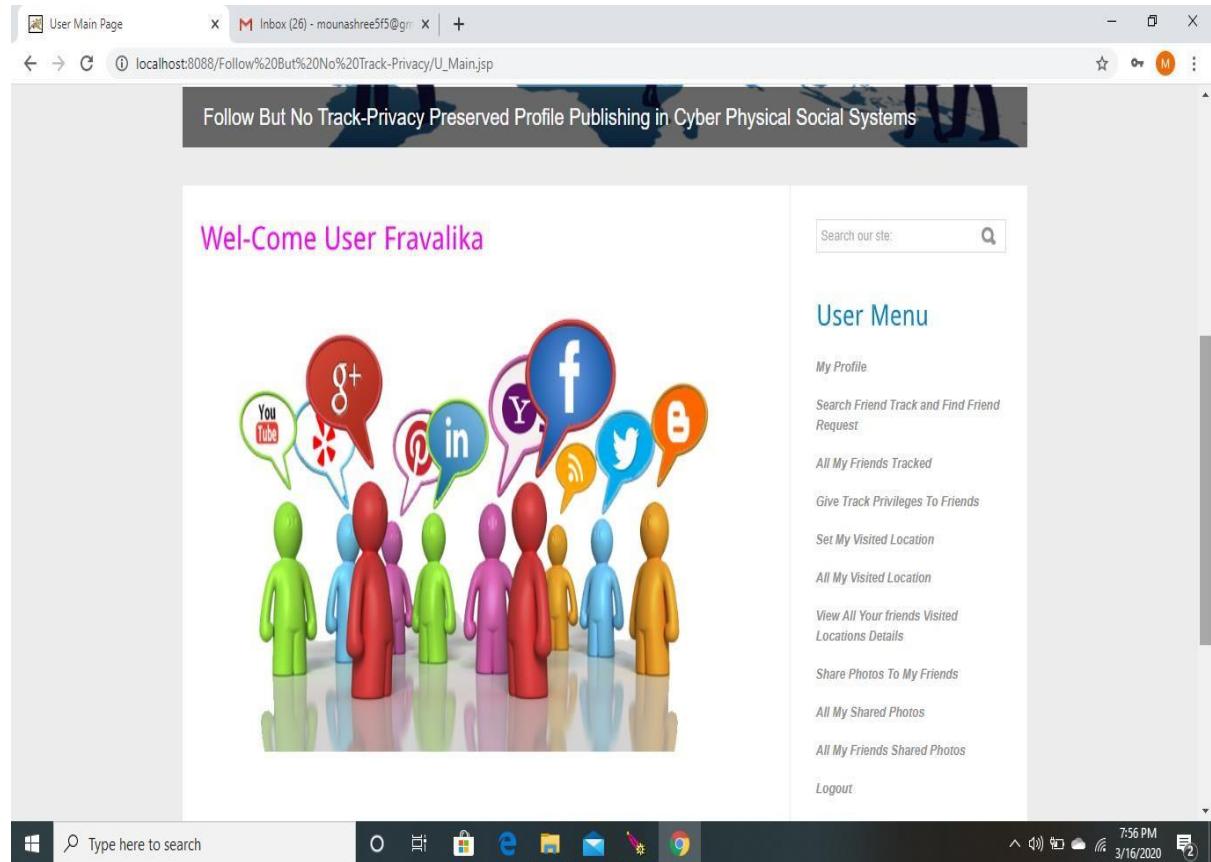


Fig. 11. User Home Page.

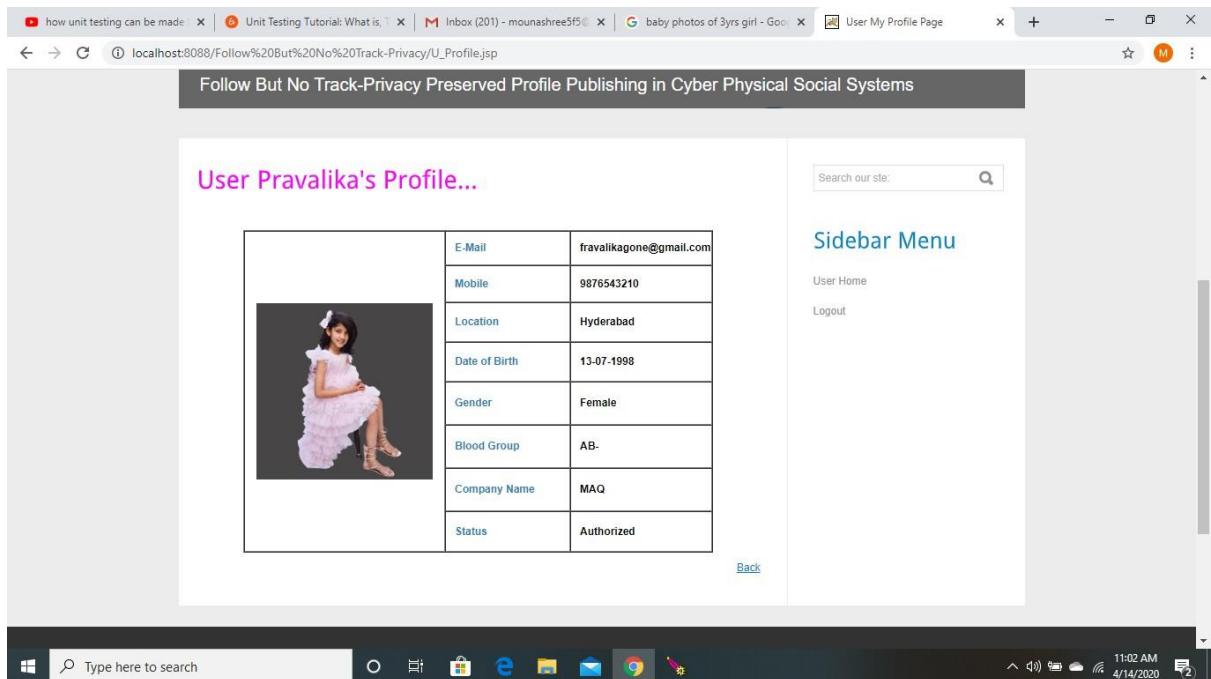


Fig. 12. User profile details.

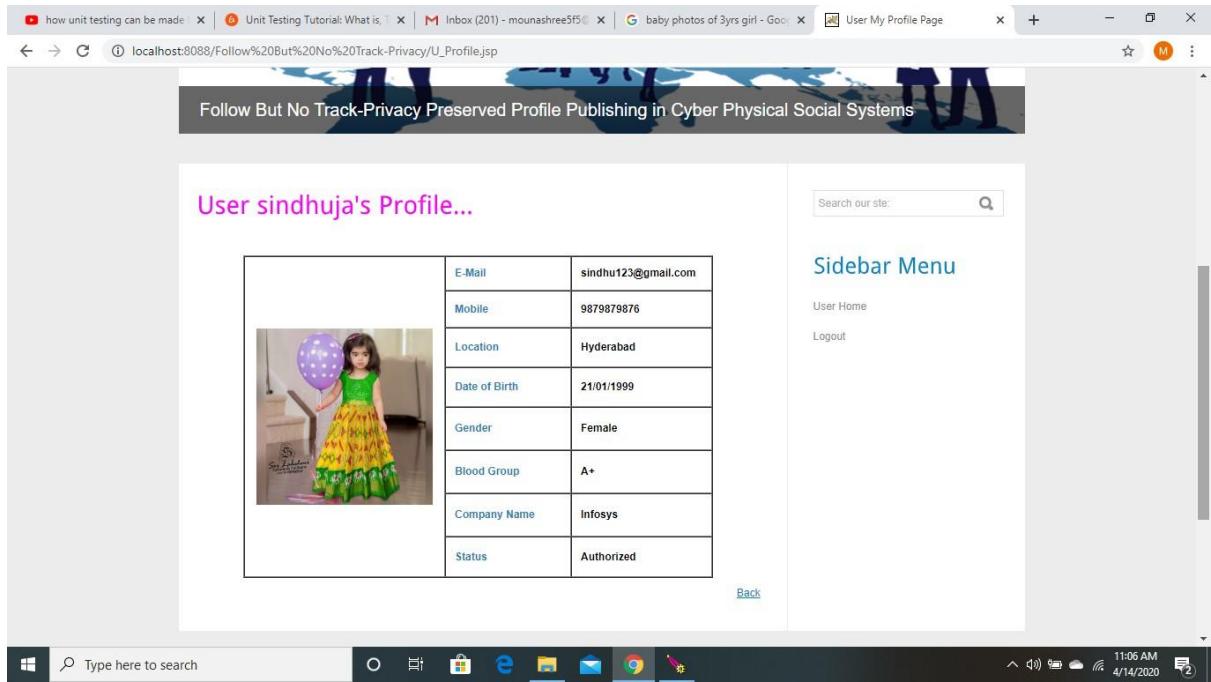


Fig. 13. User profile details.

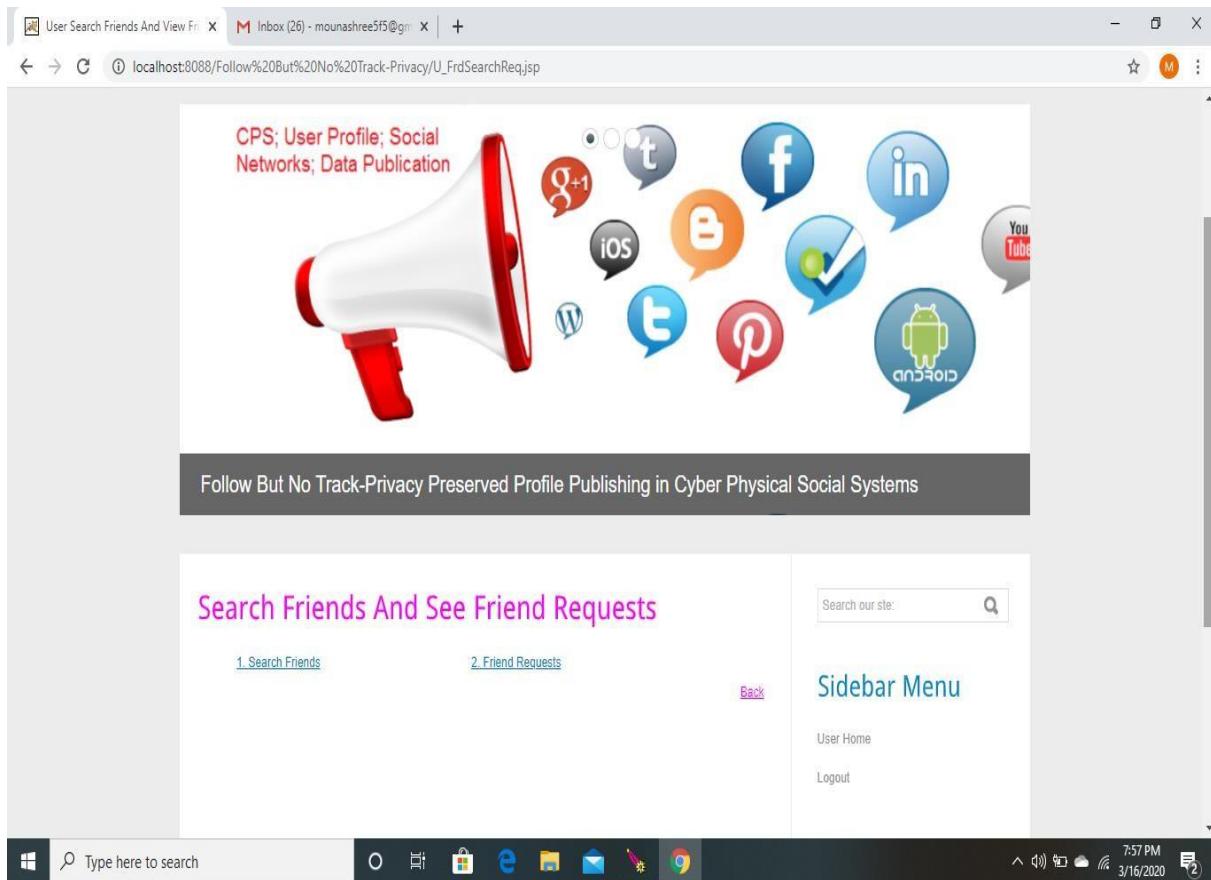


Fig. 14. Search friends and see friend requests.

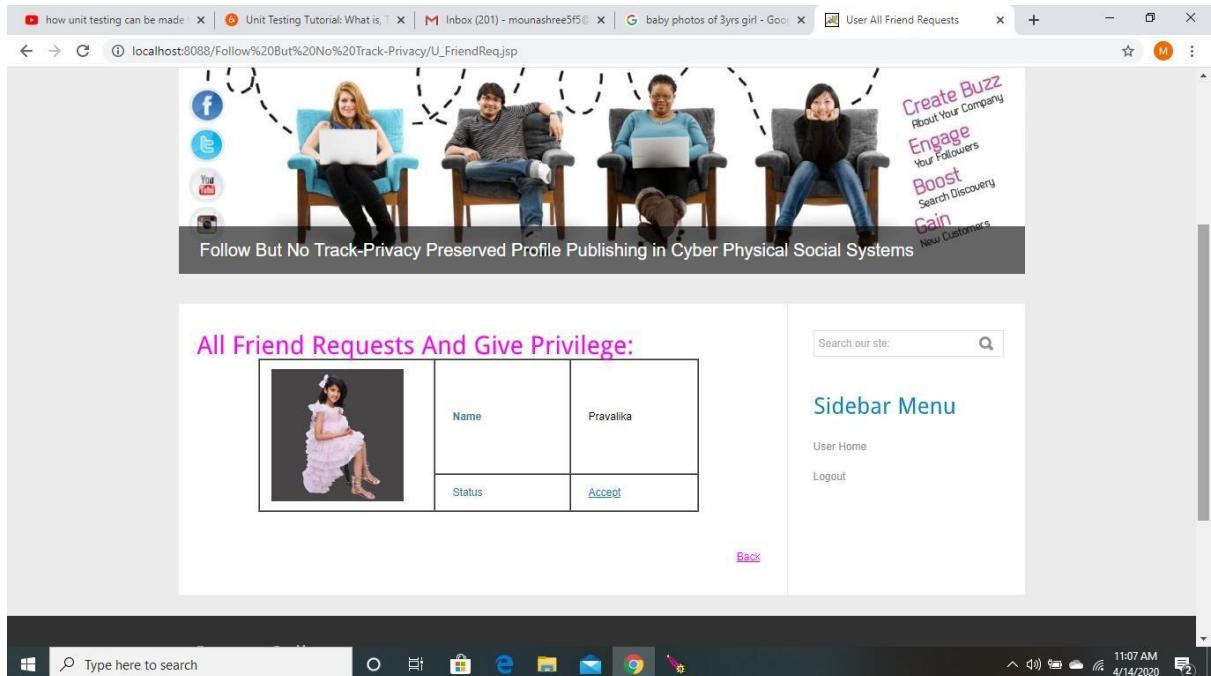


Fig. 15. All friend request and give privilege.

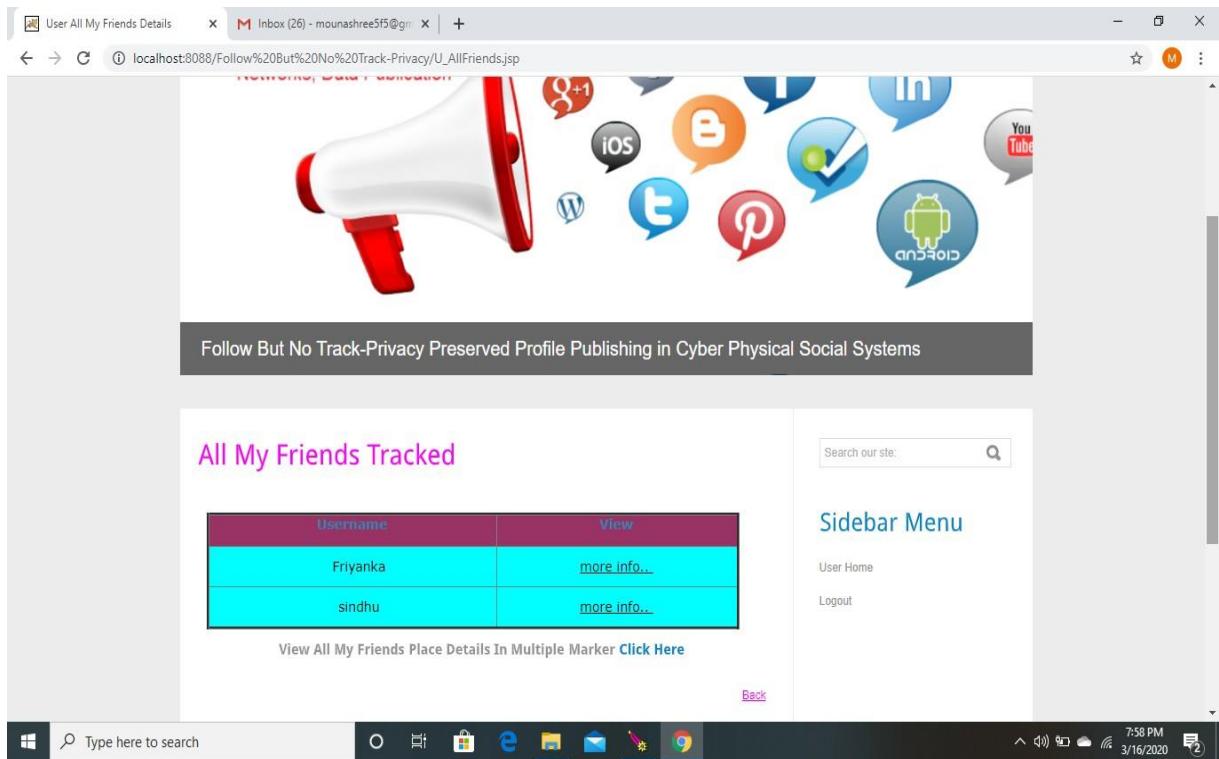


Fig. 16. All my friends tracked.

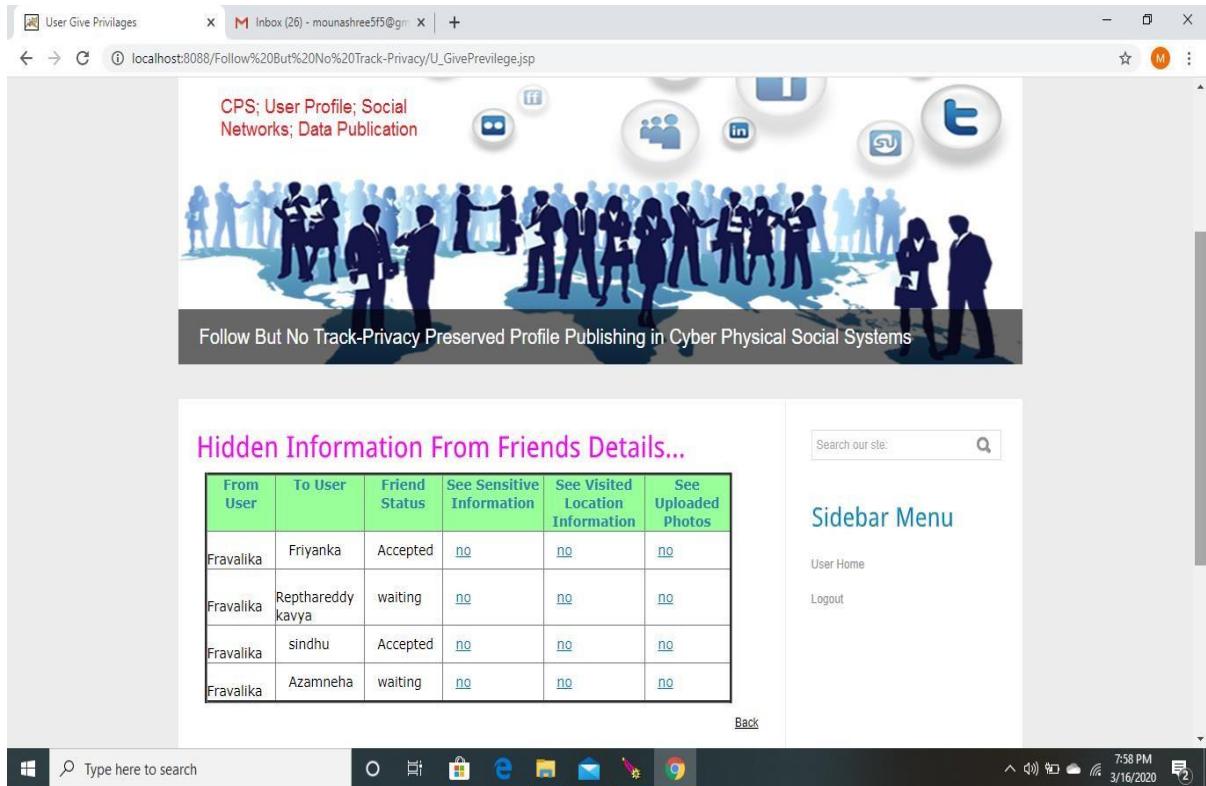


Fig. 17. Hidden information from friends details.

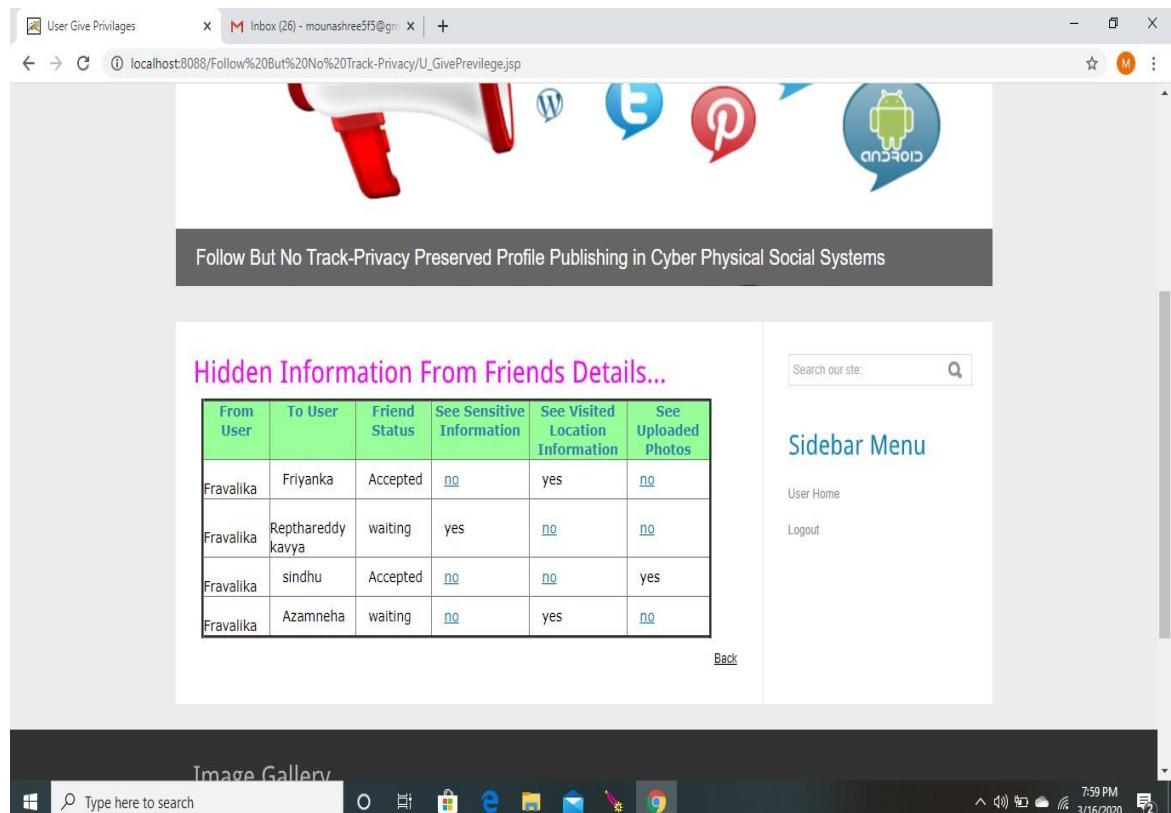


Fig. 18. Hidden information from friends details.

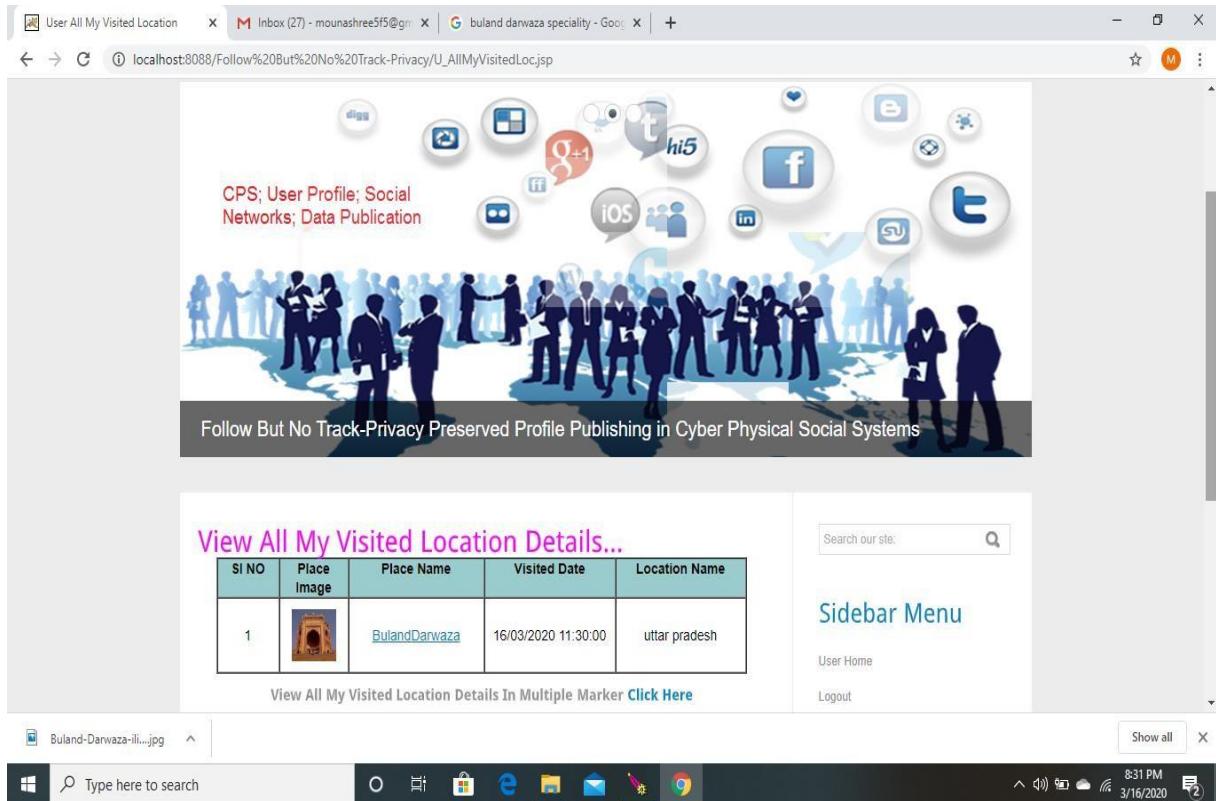


Fig. 19. View all my visited location details.

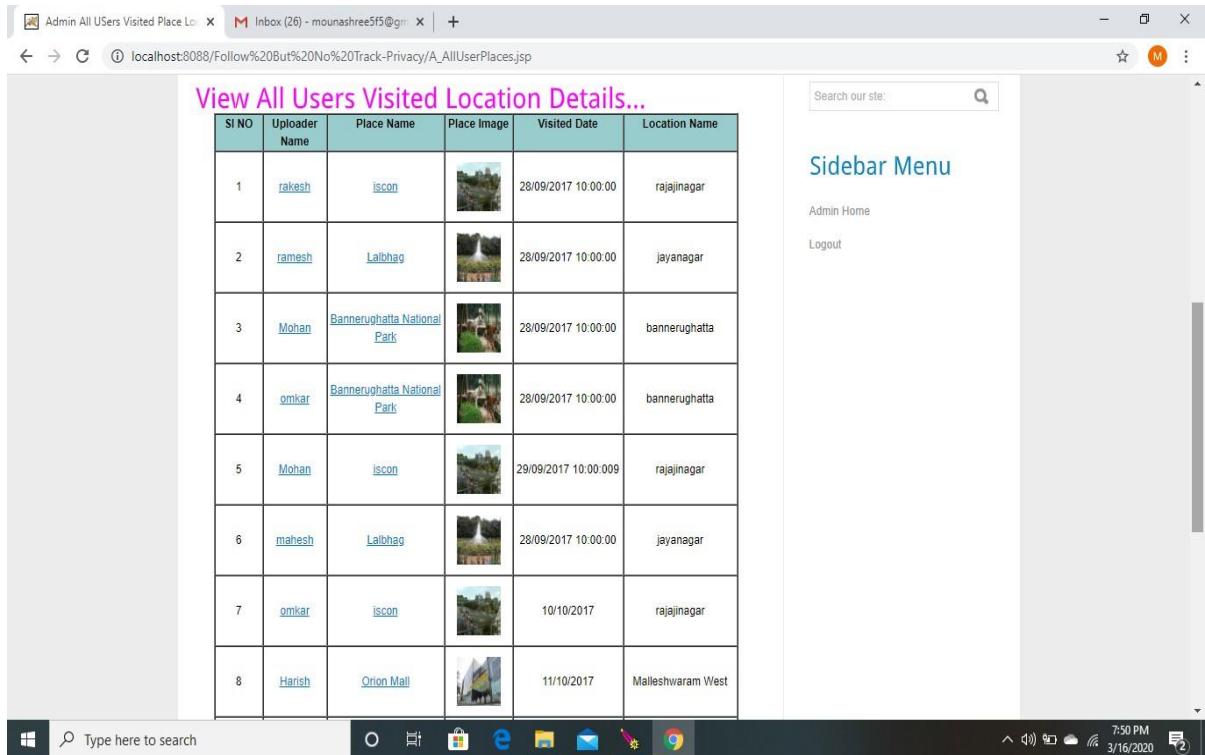


Fig. 20. View all users visited location details.

Admin Users Hidden Information | M Inbox (26) - mounashree5f5@gmail.com | +

localhost:8088/Follow%20But%20No%20Track-Privacy/A\_UserHiddenInf.jsp

Users Hidden Sensitive Information

USER NAME	HIDDEN FROM
Azamneha	sindhu
Azamneha	friyanka
Fravalika	Friyanka
Fravalika	Rephareddy kavya
Fravalika	sindhu
Fravalika	Azamneha
friyanka	Fravalika
friyanka	Azamneha
friyanka	Sindhu
mahesh	omkar
omkar	rakesh
omkar	ramesh
rakesh	omkar
ramesh	omkar
sd	Manjunath
sindhu	friyanka

Search our site:

Sidebar Menu

- Admin Home
- Logout

Windows Taskbar: Type here to search, File, Start, Store, Edge, Mail, Photos, Google Chrome, 7:52 PM, 3/16/2020

Fig. 21. Users hidden sensitive information.

Admin Users Hidden Information | M Inbox (26) - mounashree5f5@gmail.com | +

localhost:8088/Follow%20But%20No%20Track-Privacy/A\_UserHiddenInf.jsp

All Users Hidden Visited Location Details...

SI NO	Uploader Name	Hidden From	Place Name	Place Image	Visited Date And Time	Location Name
1	mahesh	omkar	Lalbhag		28/09/2017 10:00:00	jayanagar
2	omkar	rakesh	Bannerughatta National Park		28/09/2017 10:00:00	bannerughatta
3	omkar	rakesh	iscon		10/10/2017	rajajinagar
4	omkar	ramesh	Bannerughatta National Park		28/09/2017 10:00:00	bannerughatta
5	omkar	ramesh	iscon		10/10/2017	rajajinagar
6	rakesh	omkar	iscon		28/09/2017 10:00:00	rajajinagar
7	ramesh	rakesh	Lalbhag		28/09/2017 10:00:00	jayanagar
8	ramesh	omkar	Lalbhag		28/09/2017 10:00:00	jayanagar

Windows Taskbar: Type here to search, File, Start, Store, Edge, Mail, Photos, Google Chrome, 7:52 PM, 3/16/2020

Fig. 22. All users hidden visited location details.

Follow But No Track-Privacy Preserved Profile Publishing in Cyber Physical Social Systems

All Users Shared Photos Details...

Sl NO	Shared User Name	Photos	Photo Name	Uploaded Date	Photo Rank	Photo Rate	Photo Comments
1	rakesh		peacock	04/10/2017 15:50:04	9	★★★	<a href="#">Comments</a>
2	ramesh		parrot	04/10/2017 16:01:08	12	★★★	<a href="#">Comments</a>
3	rakesh		hp Laptop	06/10/2017 13:25:19	3		<a href="#">Comments</a>
4	Harish		Train Accident	11/10/2017 13:00:06	2	★★	<a href="#">Comments</a>
5	Manjunath		BMW Car	11/10/2017 16:23:18	2	★★	<a href="#">Comments</a>

Back

Fig. 23. All users shared photos details.

Follow But No Track-Privacy Preserved Profile Publishing in Cyber Physical Social Systems

All Users Hidden Added Photos Details...

Sl NO	Uploader Name	Hidden From	Image	Photo Name
1	rakesh	omkar		peacock
2	rakesh	omkar		hp Laptop
3	ramesh	rakesh		parrot
4	ramesh	omkar		parrot

Total Number Of Users Hidden Added Photos From Friends In Chart [Click Here](#)

Back

Fig. 24. All users Hidden added photos details.



Fig. 25. Number of users visited location details.

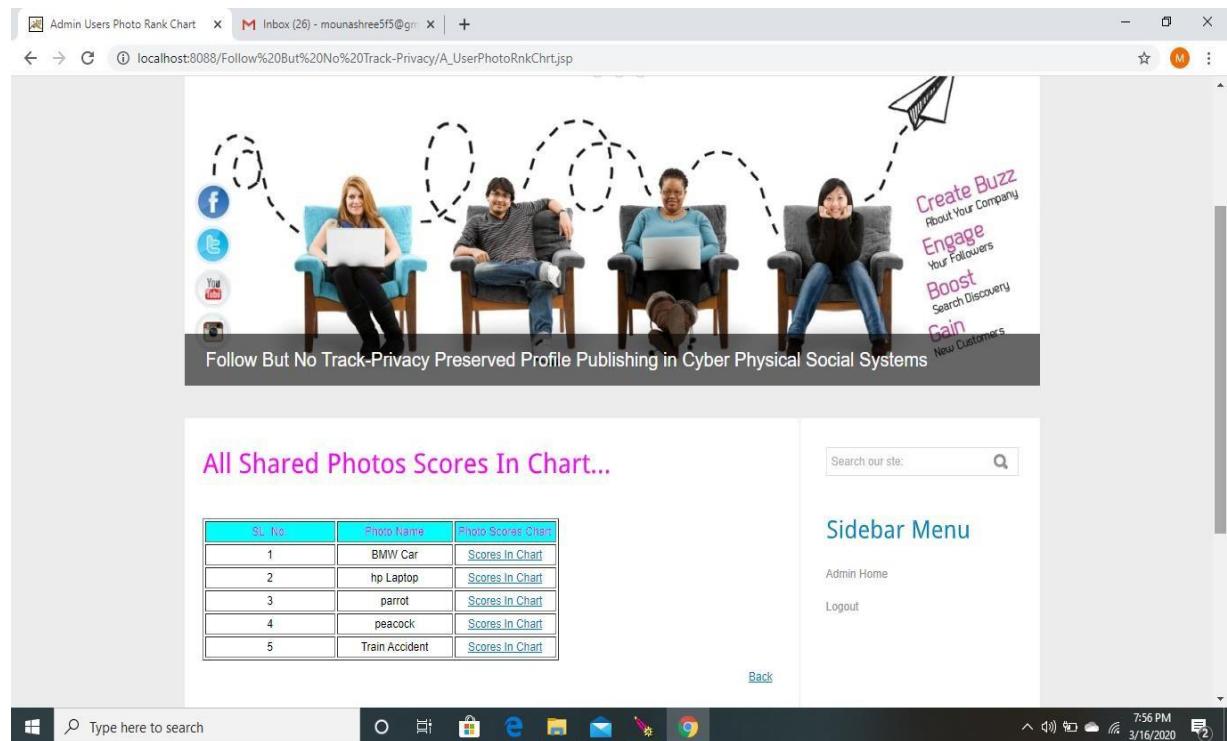


Fig. 26. All shared photos scores in chart.

## **10. CONCLUSION**

In this paper, we investigate the problem of privacy preservation in CPSSs which inherit features from both cyber-physical systems and social networks and face novel privacy issues. A thoroughly designed countermeasure is expected to handle the privacy issues for physical data, while maintaining a good utility as in social networks. We prove that the proposed algorithm can achieve a maximal number of published records. The evaluation on real dataset validates the effectiveness of the algorithm on both physical privacy and social utility. Our study could work as a general tool in preserving users' privacy when they share their physical data with others in CPSSs.

## REFERENCES

- [1] A. Sheth, P. Anantharam, and C. Henson, “Physical-cyber-social computing: An early 21st century approach,” *IEEE Intelligent Systems*, vol. 28, no. 1, pp. 78–82, 2013.
- [2] F. Li, C. Tian, T. Li, and Y. Wang, “Energy efficient social routing framework for mobile social sensing networks,” *Tsinghua Science and Technology*, vol. 21, no. 4, pp. 363–373, 2016.
- [3] T. Qiu, D. Luo, F. Xia, N. Deonauth, W. Si, and A. Tolba, “A greedy model with small world for improving the robustness of heterogeneous internet of things,” *Computer Networks*, vol. 101, pp. 127–143, 2016.
- [4] L. Hu, A. Sun, and Y. Liu, “Your neighbors affect your ratings: on geographical neighborhood influence to rating prediction,” in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 345–354, ACM, 2014.
- [5] T. Qiu, A. Zhao, R. Ma, V. Chang, F. Liu, and Z. Fu, “A task-efficient sink node based on embedded multi-core soc for internet of things,” *Future Generation Computer Systems*, 2016.
- [6] Y. Wang, D. Xu, and F. Li, “Providing location-aware location privacy protection for mobile location-based services,” *Tsinghai Science and Technology*, vol. 21, no. 3, pp. 243–259, 2016.
- [7] A. Pingley, W. Yu, N. Zhang, X. Fu, and W. Zhao, “Cap: A context aware privacy protection system for location-based services,” in *Distributed Computing Systems, 2009. ICDCS’09. 29th IEEE International Conference on*, pp. 49–57, IEEE, 2009.
- [8] K. Fawaz, H. Feng, and K. G. Shin, “Anatomization and protection of mobile apps: Location privacy threats,” in *24th USENIX Security Symposium (USENIX Security 15)*, pp. 753–768, 2015.
- [9] T. Qiu, X. Liu, L. Feng, Y. Zhou, and K. Zheng, “An efficient treebased self-organizing protocol for internet of things,” *IEEE Access*, vol. 4, pp. 3535–3546, 2016.
- [10] Z. He, Z. Cai, J. Yu, X. Wang, Y. Sun, and Y. Li, “Cost-efficient strategies for restraining rumor spreading in mobile social networks,” *IEEE Transactions on Vehicular Technology*, 2016.

- [11] E. Zheleva and L. Getoor, “To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles,” in Proceedings of the 18th international conference on World wide web, pp. 531–540, ACM, 2009.
- [12] N. Z. Gong and B. Liu, “You are who you know and how you behave: Attribute inference attacks via users’ social friends and behaviors,” in 25th USENIX Security Symposium (USENIX Security 16), (Austin, TX), pp. 979–995, USENIX Association, Aug. 2016.
- [13] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou, “Hidden voice commands,” in 25th USENIX Security Symposium (USENIX Security 16), Austin, TX, 2016.
- [14] Q. Yang, J. Yang, W. Yu, D. An, N. Zhang, and W. Zhao, “On false data-injection attacks against power system state estimation: Modeling and countermeasures,” IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 3, pp. 717–729, 2014.
- [15] H. Almuhamdi, F. Schaub, N. Sadeh, I. Adjerid, A. Acquisti, J. Gluck, L. F. Cranor, and Y. Agarwal, “Your location has been shared 5,398 times!: A field study on mobile app privacy nudging,” in Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 787–796, ACM, 2015.
- [16] Y. Xu, T. Price, J.-M. Frahm, and F. Monrose, “Virtual u: Defeating face liveness detection by building virtual models from your public photos,”
- [17] L. Zhang, Z. Cai, and X. Wang, “Fakemask: a novel privacy preserving approach for smartphones,” IEEE Transactions on Network and Service Management, vol. 13, no. 2, pp. 335–348, 2016.
- [18] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, “Optimal geoindistinguishable mechanisms for location privacy,” in Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 251–262, ACM, 2014.
- [19] B. Zhang, C. H. Liu, J. Lu, Z. Song, Z. Ren, J. Ma, and W. Wang, “Privacy-preserving qoi-aware participant coordination for mobile crowdsourcing,” Computer Networks, 2016.
- [20] X. Zheng, Z. Cai, J. Li, and H. Gao, “Location-privacy-aware review publication mechanism for local business service systems,” INFOCOM, 2017 Proceedings IEEE, 2017.