CONTENT VENUE AWARE TOPIC MODEL FOR GROUP EVENT RECOMMENDATION

A PROJECT REPORT

Submitted by

KAVIRAJ. A [REGISTER NO:211414104113]

NAKUL. R [REGISTER NO: 211417104158]

PARANTHAMAN. K [REGISTER NO:211414104178]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2021

BONAFIDE CERTIFICATE

Certified that this project report **"CONTENT VENUE AWARE TOPIC MODEL FOR GROUP EVENT RECOMMENDATION"** is the bonafide work of **"KAVIRAJ A (211417104113), NAKUL R (211417104158), PARANTHAMAN K (211417104178)"** who carried out the project work under my supervision.

SIGNATURE

Dr.S.MURUGAVALLI,M.E.,Ph.D.
PROFESSOR
HEAD OF THE DEPARTMENT

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

SIGNATURE

Mr. S. SUBBURAJ
SUPERVISOR
ASSISTANT PROFESSOR

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

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on.....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like express our heartfelt and sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHIKUMAR, M.E., Ph.D.,** and **Tmt. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my **Mr .S. SUBURAJ** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

KAVIRAJ A
NAKUL R
PARANTHAMAN K

ABSTRACT

Event-based online social platforms, such as Meetup and Plancast, have experienced increased popularity and rapid growth in recent years. In EBSN setup, selecting suitable venues for hosting events, which can attract a great turnout, is a key challenge. In this paper, we present a deep learning based venue recommendation system Deep Venue which provides context driven venue recommendations for the Meetup event-hosts to host their events. The crux of the proposed relies on the notion of similarity between multiple Meetup entities such as events, venues, groups etc. We develop deep learning techniques to compute a compact descriptor for each entity, such that two entities (say, venues) can be compared numerically. Notably, to mitigate the scarcity of venue related information in Meetup, we leverage on the cross domain knowledge transfer from popular LBSN service Yelp to extract rich venue related content. For hosting an event, the proposed Deep Venue model computes a success score for each candidate venue and ranks those venues according to the scores and finally recommend the top k venues. Our rigorous evaluation on the Meetup data collected for city Chicago shows that Deep Venue significantly outperforms the baselines algorithms. Precisely, for 84% of events, the correct hosting venue appears in the top 5 of Deep Venue recommended list.

TABLE OF CONTENTS

CHAPTER NO. **TITLE** **PAGE NO.**

LIST OF FIGURES

LIST OF ABBREVATIONS

JDK	Java Development Toolkit
DEX	Dalvik Executables
TCP	Transmission Control Protocol
IP	Internet Protocol
HTTP	Hyper Text Transfer Protocol
ADT	Android Development Tool

CHAPTER 1

INTRODUCTION

Aim:

The main aim of this project is to recommend suitable events to a single or a group of users based on users preferences and current context.

Synopsis:

Venue recommendation for hosting popular Meet up events is a perfect example of a recommendation problem where multiple entities interact and influence each other. Selecting suitable venues for hosting events is an important aspect for a venue recommendation. In this paper, we present a deep learning based venue recommendation system deep venue which provides context driven venue recommendations for the Meet up event-hosts to host their events. Identifying suitable venue from multiple options to organize a successful Meet up event (attracting large population) is essential for the event hosts. While recommending suitable events and groups to meet up users, we have taken into account the users preference of visiting venues extensively, Motivated by these endeavors, in this paper, we propose a venue recommendation methodology for Meet up hosts to organize popular events. The Recommended venue should be suitable for all the member in a group.

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

In existing system depending on the target users, the recommendation systems on EBSN can broadly be categorized into two classes (a) recommendation for general Meet up members and (b) recommendation for Meet up group organizers and event hosts. where most of the endeavors focused on recommending suitable events to a single or collection of Meet up users, few attempts have been made in recommending potential Meet up groups to the members for joining. In the context of Location Based Social Networks (LBSN), stand-alone point-of-interest (POI) recommendation recommends customers with individual venues (say restaurant or shopping mall) for future visits, depending on their past visits and preferences.

Problem Definition:

- ✔ The venue recommender needs to rank all the venues according to their abilities of making the event successful.
- ✔ The key challenge in developing deep venue model, which involves multiple sources of similarities, is dealing with the heterogeneity present in the input entities.

2.2 PROPOSED SYSTEM

Our proposed system focus on recommending venue for a group or single user. In this system users can add their past and future events successfully hosted in their venue and also the users who attended the events can also add the event details by doing this we can avoid data sacristy. While searching for a venue based on event the end user has to select type of recommendation whether it is a single or group recommendation. For single user recommendation user can select their preference that user wishes to attend the events based on the user preference past and present events available for user nearby places will be displayed. In group recommendation end user can select list of people going to participate in an event, location of all the selected members will be collected and center point of location gathered calculated and based on user preference venue will be recommended user can select the venue and send place details to the group the route map for the location will be displayed for the users.

Advantages:

- ✔ In group recommendation venue suitable for the entire user in the group will be recommended so the users can attend event without difficulties.
- ✔ User can select the events based on their preference whether the environment of place should be noisy, musical or pleasant.

CHAPTER 3

REQUIREMENT SPECIFICATIONS

3.1 INTRODUCTION

Meetup is an event based social networking (EBSN) portal for hosting events in various localities around the world [1]. Event recommendation is an important problem in Meetup [1], [2], [3].Prior studies recommend suitable events to a single or a group of users based on users' past preferences and current context [1], [2], [3] considering factors such as users' profile, location and social features [4]. However, only few attempts have been made to provide guidance to the Meetup hosts for organizing popular events. For instance, portals like 'Peerspace'2 display theavailability of venues to host a few specific types of events (say, Corporate Dinner, Photo Shoot, Retreat, Wedding etc); however, they do not provide any intelligent recommendation. Identifying suitable venue from multiple options to organize a successful Meetup event (attracting large population) is essential for the event hosts. For instance, a technical talk can be hosted in an auditorium as well as in a cafeteria depending on the size and type of the event (see Fig. 1(a)). In [5] Liu et al. demonstrate a sharp decrease in event participation with increasing distance between attendees and the event venue. While recommending suitable events [1], [2] and groups [6] to Meetup users, researchers have taken into account the users' preference of visiting venues extensively. Motivated by these endeavors, in this paper, we propose a venue recommendation methodology for Meetup hosts to organize popular events. Recommending suitable venues involves multiple challenges. 'Data sparsity' is a major challenge. In general, most user-item recommender systems [7], [8] learn user preferences based on past user-item interaction histories. Unique to the venue recommendation problem is the lack of history for a particular event.Secondly, a widely adopted approach to mitigate data sparsity in user-item recommendation is to use additional metadata regarding users

& items. For instance, in the of Point-of-Interest (POI) recommendations in Location Based Social Network (LBSN), recent studies [9], [10] have leveraged on content information of prior visited locations to infer user interests. Unfortunately, Meetup does not provide any additional semantic information associated with venues. Third, the popularity of an event also depends on the preference of the group members. For instance, two Meetup groups 'Chicago Bar Trivia Group' and 'The Fun Chicago Single' have availed 'Bull and Bear' restaurant in Chicago to host their events. In case of the former one, all their events at 'Bull and Bear' have attracted more than 30% members whereas for the later one attendance is below 5% for all the events hosted at the same restaurant. Finally, the choice of the venues for hosting successful Meetup events depend on the event specific factors. For instance, a thoughtful conversation requires location where the noise level is low, whereas a corporate presentation needs a space with high quality audio-visual facility. In a nutshell, the suitability of a specific venue for hosting an event depends on the complex interactions across multiple entities such as groups, events etc.

3.2 HARDWARE AND SOFTWARE SPECIFICATION

3.2.1 HARDWARE REQUIREMENTS

- Hard Disk : 500GB and Above
- RAM : 4 GB and Above
- Processor : I3 and Above

3.2.2 SOFTWARE REQUIREMENTS

- Windows 7 and above

- JDK 1.7
- Xampp
- Android Studio 3.4
- Android Phone

3.3 TECHNOLOGIES USED

- JAVA
- Android

3.3.1 JAVA

Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. The language, initially called Oak (named after the oak trees outside Gosling's office), was intended to replace C++, although the feature set better resembles that of Objective C.

3.3.1.1 INTRODUCTION TO JAVA

Java has been around since 1991, developed by a small team of Sun Microsystems developers in a project originally called the Green project. The intent of the project was to develop a platform-independent software technology that would be used in the consumer electronics industry. The language that the team created was originally called Oak.

The first implementation of Oak was in a PDA-type device called Star Seven (*7) that consisted of the Oak language, an operating system called GreenOS, a user interface, and hardware. The name *7 was derived from the telephone sequence that was used in the team's office and that was dialed in order

to answer any ringing telephone from any other phone in the office.

Around the time the First Person project was floundering in consumer electronics, a new craze was gaining momentum in America; the craze was called "Web surfing." The World Wide Web, a name applied to the Internet's millions of linked HTML documents was suddenly becoming popular for use by the masses. The reason for this was the introduction of a graphical Web browser called Mosaic, developed by ncSA. The browser simplified Web browsing by combining text and graphics into a single interface to eliminate the need for users to learn many confusing UNIX and DOS commands. Navigating around the Web was much easier using Mosaic.

It has only been since 1994 that Oak technology has been applied to the Web. In 1994, two Sun developers created the first version of Hot Java, and then called Web Runner, which is a graphical browser for the Web that exists today. The browser was coded entirely in the Oak language, by this time called Java. Soon after, the Java compiler was rewritten in the Java language from its original C code, thus proving that Java could be used effectively as an application language. Sun introduced Java in May 1995 at the Sun World 95 convention.

Web surfing has become an enormously popular practice among millions of computer users. Until Java, however, the content of information on the Internet has been a bland series of HTML documents. Web users are hungry for applications that are interactive, that users can execute no matter what hardware or software platform they are using, and that travel across heterogeneous networks and do not spread viruses to their computers. Java can create such applications.

3.3.1.2 WORKING OF JAVA

For those who are new to object-oriented programming, the concept of a class will be new to you. Simplistically, a class is the definition for a segment of code that can contain both data (called attributes) and functions (called methods).

When the interpreter executes a class, it looks for a particular method by the name of **main,** which will sound familiar to C programmers. The main method is passed as a parameter an array of strings (similar to the argv [] of C), and is declared as a static method.

To output text from the program, we execute the **println** method of **System.out,** which is java's output stream. UNIX users will appreciate the theory behind such a stream, as it is actually standard output. For those who are instead used to the Wintel platform, it will write the string passed to it to the user's program.

Java consists of two things :

- Programming language
- Platform

3.3.1.3 THE JAVA PROGRAMMING LANGUAGE

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

- Simple
- Object-oriented
- Distributed
- Interpreted

- Robust
- Secure

- Architecture-neutral
- Portable
- High-performance
- Multithreaded
- Dynamic

The code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Java is unusual in that each Java program is both co implied and interpreted. With a compiler, you translate a Java program into an intermediate language called **Java byte codes** – the platform independent codes interpreted by the Java interpreter. With an interpreter, each Java byte code instruction is parsed and run on the computer. Compilation happens just once; interpretation occurs each time the program is executed. This figure illustrates how it works:

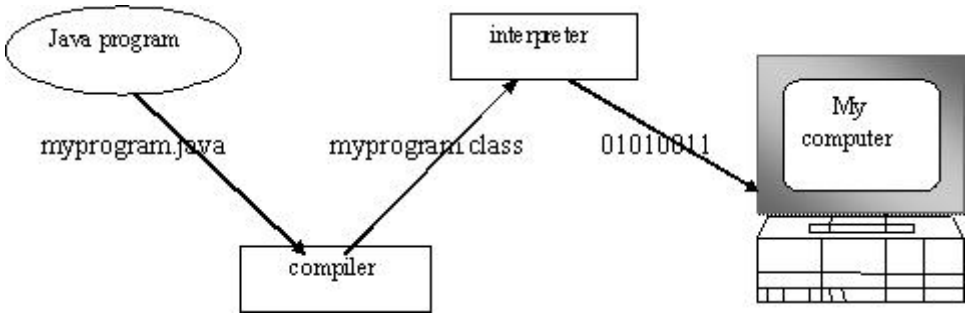
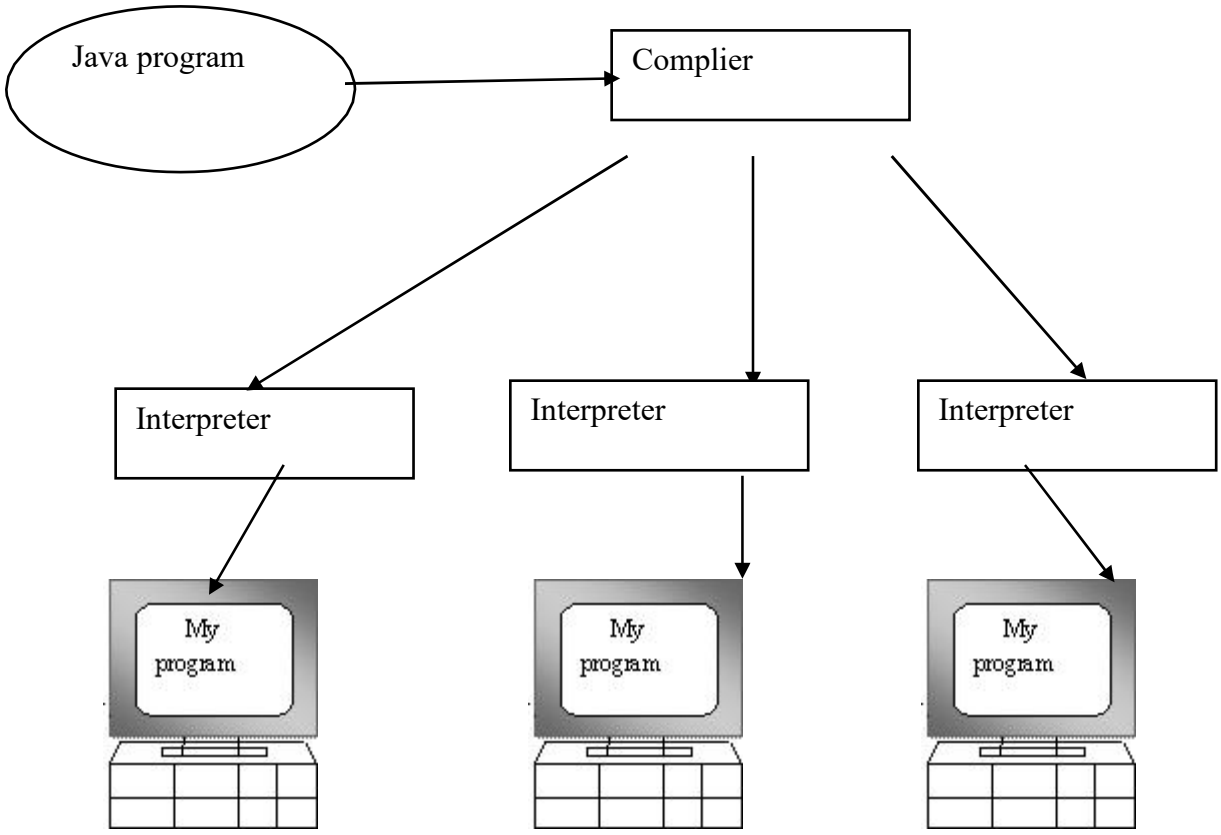


Fig.3.1

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

make “write once, run anywhere” possible.

You can compile your Java program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the JVM. For example, that same Java program can e run on Windows NT, Solaris and Macintos



3.3.1.4 THE JAVA PLATFORM

A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's a software-

only platform that runs on top of other, hardware-based platforms. Most other platforms are described as a combination of hardware and operating system.
The Java platform has two components :

- The Java Virtual Machine (JVM)
- The Java Application Programming Interface (Java API)

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

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries **(packages)** of related components. The following figure depicts a Java program, such as an application or applet, that's running on the Java platform. As the figure shows, the Java API and Virtual Machine insulates the Java program from hardware dependencies.

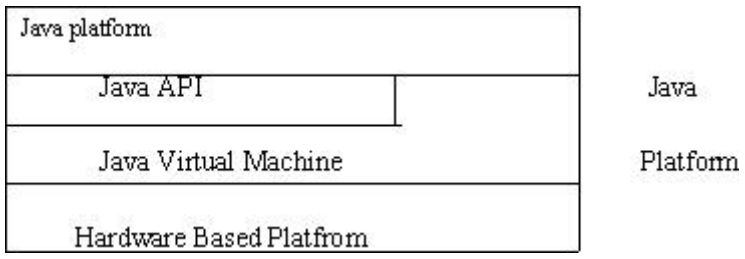


Fig.3.3

As a platform-independent environment, Java can be a bit slower than native code. However, smart compliers, weel-tuned interpreters, and just-in-time byte compilers can bring Java's performance close to that of native code without

threatening portability.

3.3.2 APACHE TOMCAT SERVER

Apache Tomcat (formerly under the Apache Jakarta Project; Tomcat is now a top level project) is a web container developed at the Apache Software Foundation. Tomcat implements the servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted. Because Tomcat includes its own HTTP server internally, it is also considered a standalone web server.

Environment

Tomcat is a web server that supports servlets and JSPs. Tomcat comes with the Jasper compiler that compiles JSPs into servlets.

The Tomcat servlet engine is often used in combination with an Apache web server or other web servers. Tomcat can also function as an independent web server. Earlier in its development, the perception existed that standalone Tomcat was only suitable for development environments and other environments with minimal requirements for speed and transaction handling. However, that perception no longer exists; Tomcat is increasingly used as a standalone web server in high-traffic, high-availability environments.

Since its developers wrote Tomcat in Java, it runs on any operating system that has a JVM.

Product features

Tomcat 3.x (initial release)

- implements the Servlet 2.2 and JSP 1.1 specifications
- servlet reloading
- basic HTTP functionality Tomcat 4.x
- implements the Servlet 2.3 and JSP 1.2 specifications
- servlet container redesigned as Catalina
- JSP engine redesigned as Jasper
- Coyote connector
- Java Management Extensions (JMX), JSP and Struts-based administration
- Tomcat 5.x
- implements the Servlet 2.4 and JSP 2.0 specifications
- reduced garbage collection, improved performance and scalability
- native Windows and Unix wrappers for platform integration
- faster JSP paring

History

Tomcat started off as a servlet specification implementation by James Duncan Davidson, a software architect at Sun. He later helped make the project open source and played a key role in its donation by Sun to the Apache Software Foundation.

Davidson had initially hoped that the project would become open-sourced and, since most open-source projects had O'Reilly books associated with them featuring an animal on the cover, he wanted to name the project after an animal. He came up with Tomcat since he reasoned the animal represented something that could take care of and fend for itself. His wish to see an animal cover eventually came true when O'Reilly published their Tomcat book with a tomcat on the cover.

3.3.3 Android Introduction:

Android is a Linux based operating system it is designed primarily for touch screen mobile devices such as smart phones and tablet computers. The operating system has developed a lot in last 15 years starting from black and white phones to recent smart phones or mini computers. One of the most widely used mobile OS these days is android. The android is software that was founded in Palo Alto of California in 2003.

The android is a powerful operating system and it supports large number of applications in Smartphones. These applications are more comfortable and advanced for the users. The hardware that supports android software is based on ARM architecture platform. The android is an open source operating system means that it's free and any one can use it. The android has got millions of apps available that can help you managing your life one or other way and it is available low cost in market at that reasons android is very popular.

The android development supports with the full java programming language. Even other packages that are API and JSE are not supported. The first version 1.0 of android development kit (SDK) was released in 2008 and latest updated version is jelly bean.

The android is a operating system and is a stack of software components which is divided into five sections and four main layers that is

3.3.3.1 Linux kernel:

The android uses the powerful Linux kernel and it supports wide range of hardware drivers. The kernel is the heart of the operating system that manages input and output requests from software. This provides basic system functionalities like process management, memory management, device management like camera, keypad, display etc the kernel handles all the things. The Linux is really good at networking and it is not necessary to interface it to the peripheral hardware. The kernel itself does not interact directly with the user but rather interacts with the shell and other programs as well as with the hard ware devices on the system.

3.3.3.2 Libraries:

The on top of a Linux kennel there is a set of libraries including open source web browser such as webkit, library libc. These libraries are used to play and record audio and video. The SQLite is a data base which is useful for storage and sharing of application data. The SSL libraries are responsible for internet security etc.

3.3.3.3 Android Runtime:

The android runtime provides a key component called Dalvik Virtual Machine which is a kind of java virtual machine. It is specially designed and optimized for android. The Dalvik VM is the process virtual machine in the android operating system. It is software that runs apps on android devices.

The Dalvik VM makes use of Linux core features like memory management and multithreading which is in a java language. The Dalvik VM enables every android application to run it own process. The Dalvik VM executes the files in the .dex format.

3.3.3.4 Application frame work:

The application frame work layer provides many higher level services to applications such as windows manager, view system, package manager, resource manager etc. The application developers are allowed to make use of these services in their application.

3.3.3.5 Applications:

You will find all the android applications at the top layer and you will write your application and install on this layer. Examples of such applications are contacts, books, browsers, services etc. Each application performs a different role in the overall applications.

Advantages:

- Android is Linux based open source operating system , it can be developed by any one
- Easy access to the android apps
- You can replace the battery and mass storage, disk drive and UDB option
- Its supports all Google services
- The operating system is able to inform you of a new SMS and Emails or latest updates.
- It supports Multitasking
- Android phone can also function as a router to share internet
- Its free to customize
- Can install a modified ROM
- Its supports 2D and 3D graphics

CHAPTER 4

Project Purpose and Scope

4.1 Purpose

The main aim of this project is to recommend suitable events to a single or a group of users based on users preferences and current context.

4.2 Project Scope

Our proposed system focus on recommending venue for a group or single user. In this system users can add their past and future events successfully hosted in their venue and also the users who attended the events can also add the event details by doing this we can avoid data sacristy. While searching for a venue based on event the end user has to select type of recommendation whether it is a single or group recommendation. For single user recommendation user can select their preference that user wishes to attend the events based on the user preference past and present events available for user nearby places will be displayed. In group recommendation end user can select list of people going to participate in an event, location of all the selected members will be collected and center point of location gathered calculated and based on user preference venue will be recommended user can select the venue and send place details to the group the route map for the location will be displayed for the users.

4.3 Product Perspective

Venue recommendation for hosting popular Meet up events is a perfect example of a recommendation problem where multiple entities interact and influence each other. Selecting suitable venues for hosting events is an important aspect for a venue recommendation. In this paper, we present a deep learning based venue recommendation system deep venue which provides context driven venue recommendations for the Meet up event-hosts to host their events. Identifying suitable venue from multiple options to organize a successful Meet up event (attracting large population) is essential for the event hosts. While recommending suitable events and groups to Meet up users, we have taken into account the users preference of visiting venues extensively, Motivated by these endeavors, in this paper, we propose a venue recommendation methodology for Meet up hosts to organize popular events. The Recommended venue should be suitable for all the member in a group.

4.4 System Features

Recommendation engine (DeepVenue) which recommends a ranked list of venues for hosting a target event. To the best of our knowledge, this is the first attempt involving deep learning for handling multi-entity interactions. The functionality of DeepVenue is based on a simple idea. A Meetup event can be successfully hosted by a group at a particular venue if (a) similar type of events have been recently hosted by similar type of groups at the same venue successfully (b) the venue is similar to the type of venues where similar type of events have been recently hosted successfully. Evidently, the proposed model relies on the notion of similarity between multiple entities (events, venues, groups etc.). However, each of these entities carry multiple complex perspectives, which makes

its representation challenging. For instance, representation of a venue includes reviews and tips posted by the clients for that venue, services and facilities available at that venue (say WiFi, garage, parking etc.), type & category of the venue (say 'Restaurants', 'Nightlife', 'Shopping', 'Hotels', 'Mass Media' etc.) etc. Hence, our first step is to develop a compact descriptor for each entity, such that two entities (say, venues) can be compared numerically. We develop a deep learning based framework to compute compact descriptors for Meetup entities through embedding Moreover, Meetup does not provide the API necessary for collecting the requisite venue details (except locational information). Close inspection reveals that in Location Based Social Networks (LBSN) such as Yelp, FourSquare etc., people provide substantial volume of information about the venues. We leverage on the cross domain knowledge transfer from popular LBSN services to extract rich venue related content for supplementing the Meetup venue details.

4.5 Design and Implementation Constraints

4.5.1 Constraints in Analysis

- Constraints as Informal Text
- Constraints as Operational Restrictions
- Constraints Integrated in Existing Model Concepts
- Constraints as a Separate Concept
- Constraints Implied by the Model Structure

4.5.2 Constraints in Design

- Determination of the Involved Classes
- Determination of the Involved Objects
- Determination of the Involved Actions
- Determination of the Require Clauses
- Global actions and Constraint Realization

4.5.3 Constraints in Implementation

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object oriented methods.

4.6 Other Nonfunctional Requirements

4.6.1 Performance Requirements

The application at this side controls and communicates with the following three main general components.

- embedded browser in charge of the navigation and accessing to the web service;
- Server Tier: The server side contains the main parts of the functionality of

the proposed architecture. The components at this tier are the following.

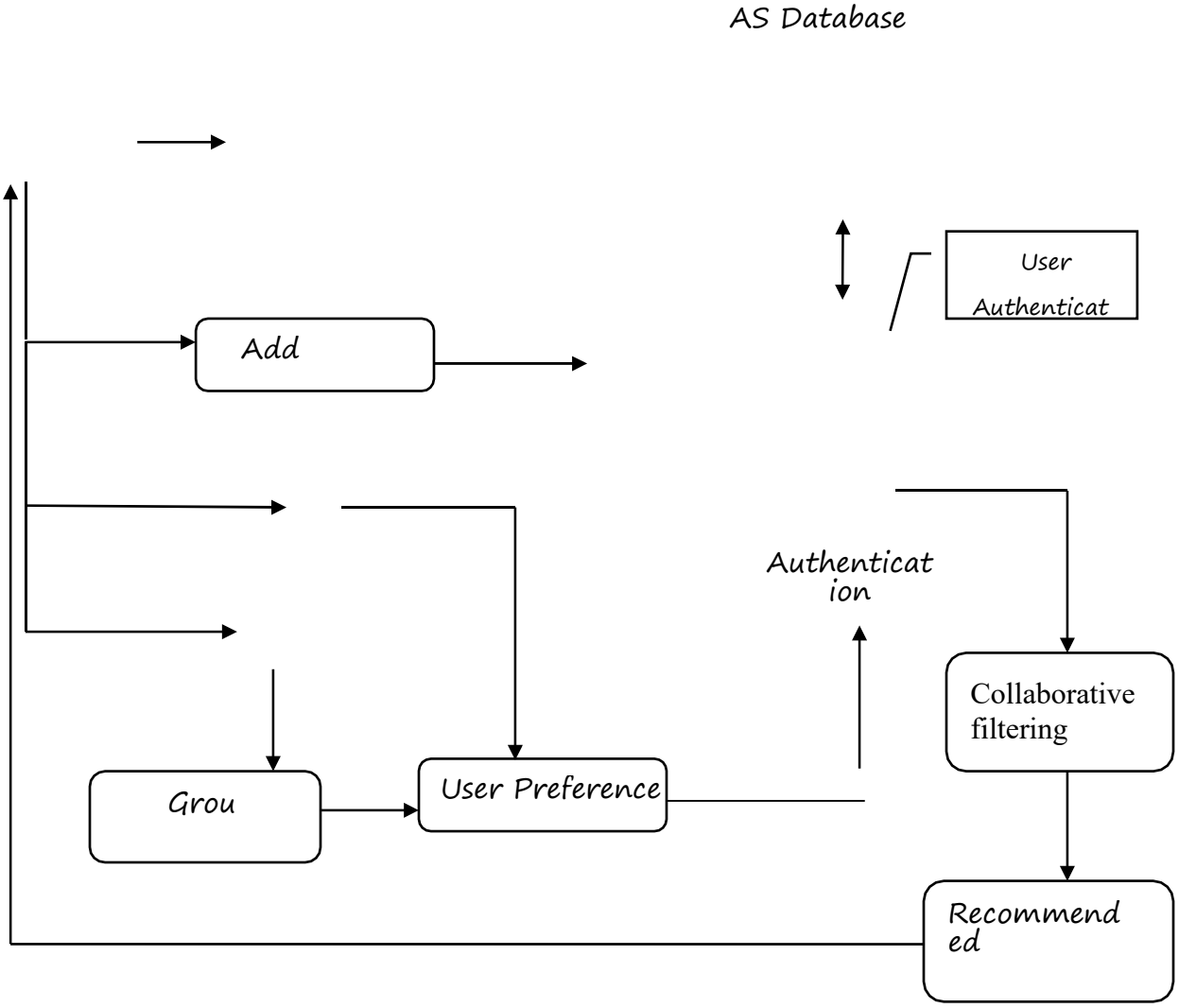
Web Server, Security Module, Server-Side Capturing Engine, Preprocessing Engine, Database System, Verification Engine, Output Module.

4.6.2 Safety Requirements

1. The software may be safety-critical. If so, there are issues associated with its integrity level
2. The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.
3. If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.
4. There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable.
5. If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.
6. Systems with different requirements for safety levels must be separated.
7. Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

CHAPTER 5

5.1 Architecture Diagram:

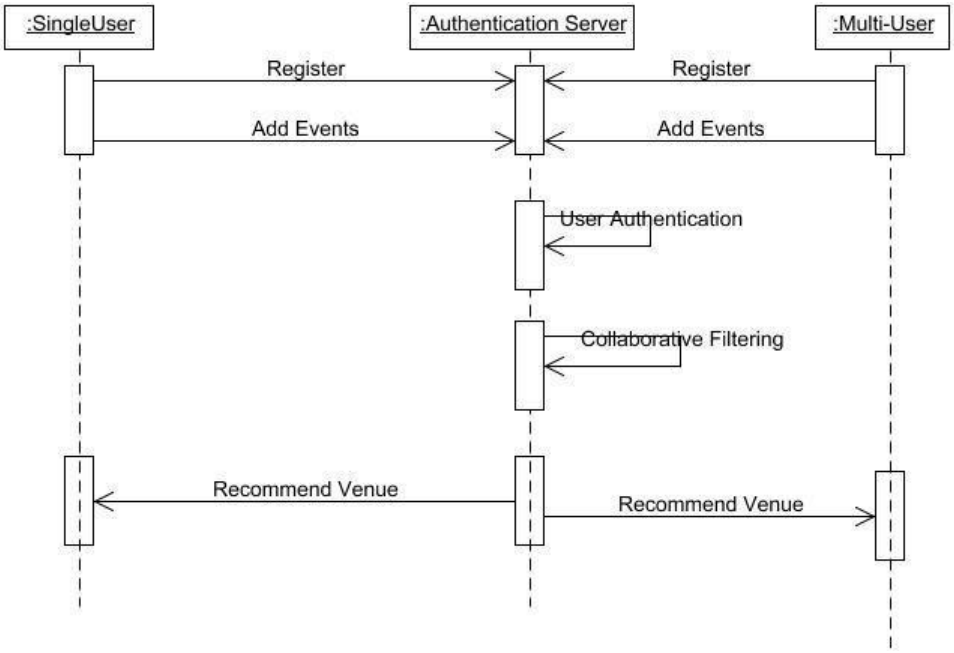


User Registration

Fig: 5.1

5.2 Sequence Diagram:

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.



5.3 UNIFIED MODELING LANGUAGE:

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

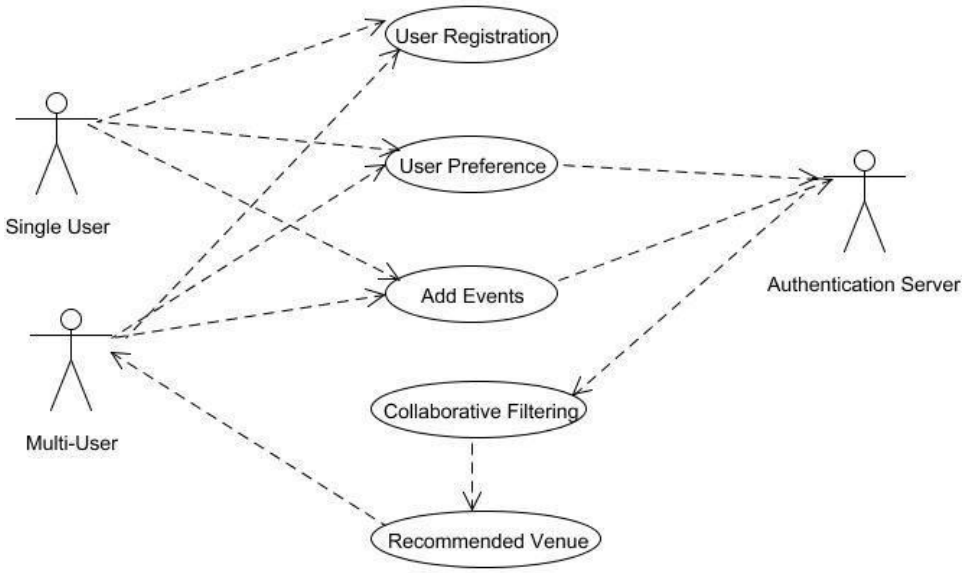
5.3 Usecase Diagram

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

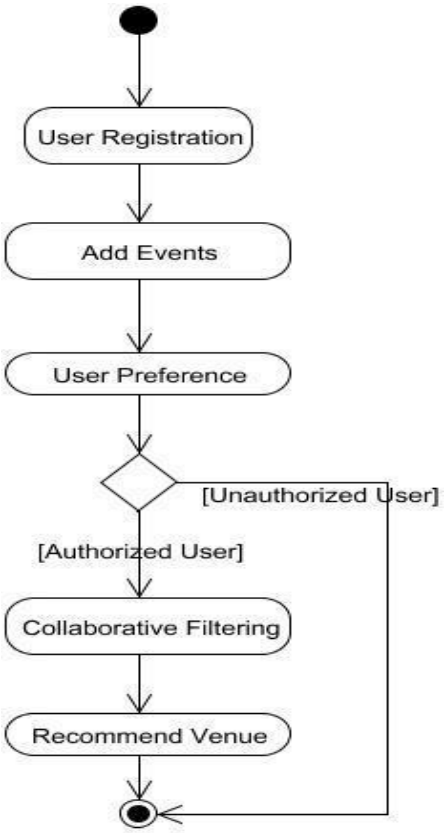


5.4 Activity Diagram:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

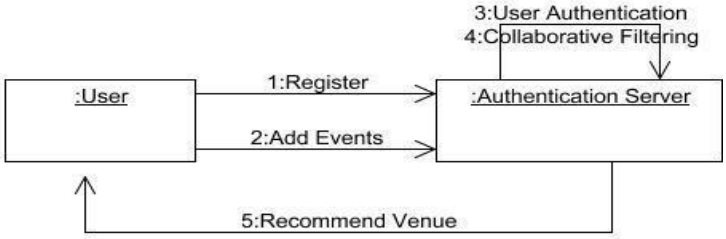
The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.



5.5 Collaboration Diagram:

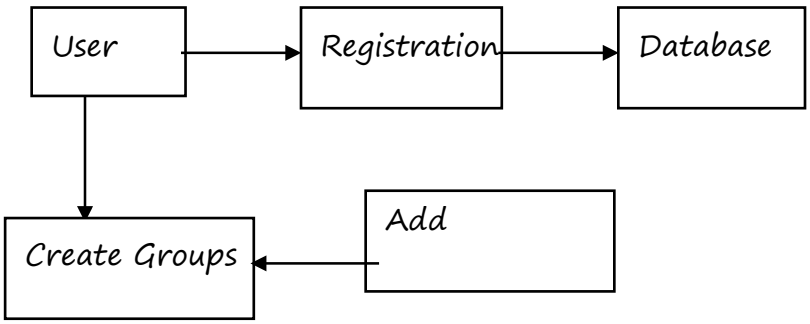
UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.



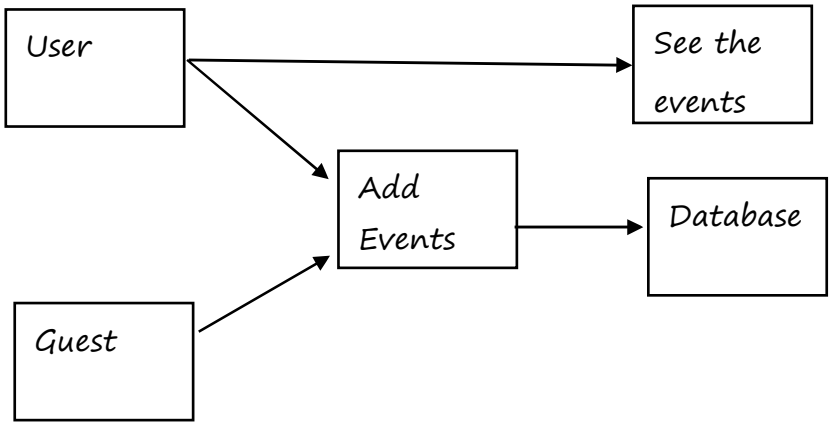
5.6 Data Flow Diagram:

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its aspects. It is a preliminary step used to create an overview of the system which can later be elaborated DFDs can also be used for visualization of data processing.

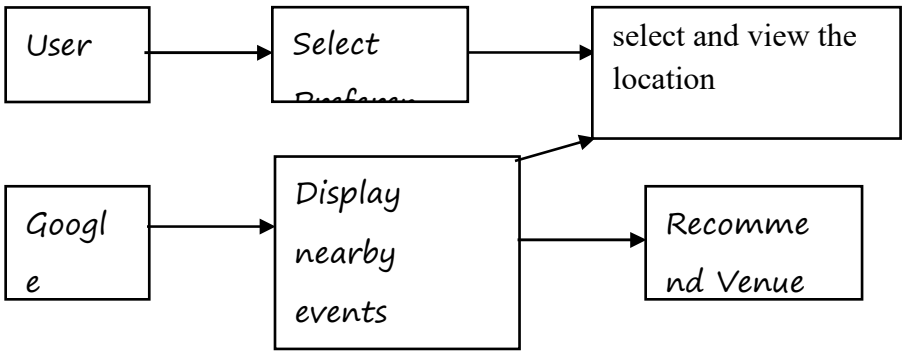
Level 0:



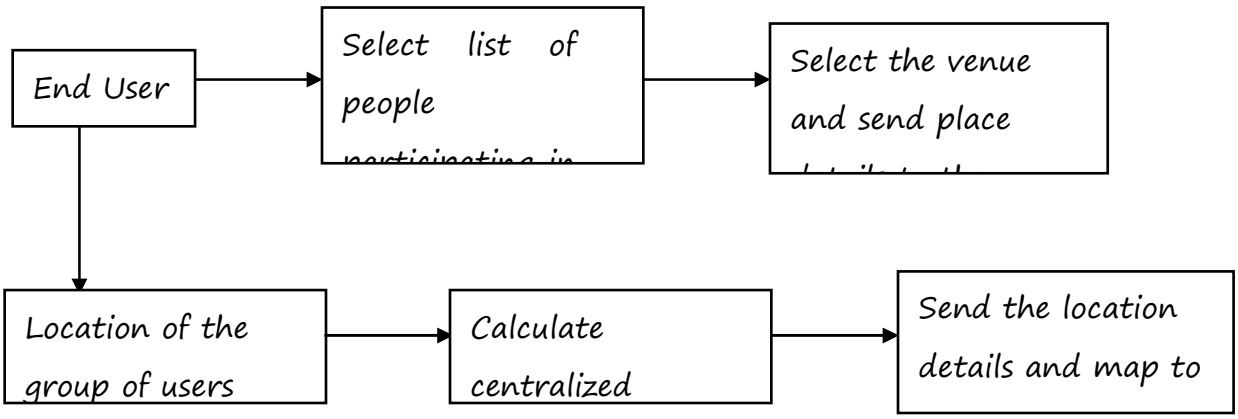
Level 1:



Level 2:

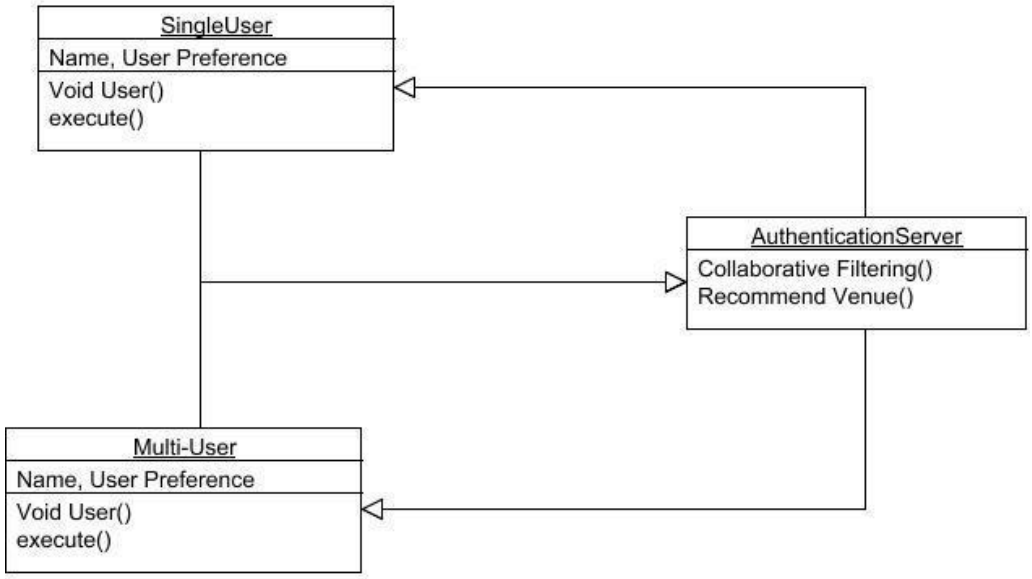


Level 3:



5.7 Class Diagram

A Class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.



CHAPTER 6

SYSTEM DESIGN

User has an initial level Registration Process. The users provide their own personal information for this process. The server in turn stores the information in its database and user can create a group to add people from their contact list of people in their list. Proprietor can add the events hosted in their venue and also the guest Attended the event can add event in database, user hosting events in their house can also select and add event detail, User can see list of events hosted within 2km radius from their location. User can select their preference on which type of place they wish to attend an event, All the nearby events which suits the user preference will be displayed in Google map User can select and view the location and event hosted in that venue based on the user preference the venues are recommended. In group recommendation end user can select list of people going to participate in an event, location of all the selected members will be collected and center point of location gathered calculated and based on user preference venue will be recommended user can select the venue and send place details to the group the route map for the location will be displayed for the users.

6.1 MODULES

- User Authentication & Create Groups
- Add Events and View nearby Events
- Single User Recommendation
- Group Recommendation

Modules Description:

User Authentication & Create Groups:

User has an initial level Registration Process. The users provide their own personal information for this process. The server in turn stores the information in its database and user can create a group to add people from their contact list of people in their list.

Add Events and View Nearby Events:

Proprietor can add the events hosted in their venue and also the guest Attended the event can add event in database, user hosting events in their house can also select and add event detail, User can see list of events hosted within 2km radius from their location.

Single User Recommendation:

User can select their preference on which type of place they wish to attend an event, All the nearby events which suits the user preference will be displayed in Google map User can select and view the location and event hosted in that venue based on the user preference the venues are recommended

Group Recommendation:

In group recommendation end user can select list of people going to participate in an event, location of all the selected members will be collected and center point of location gathered calculated and based on user preference venue will be recommended user can select the venue and send place details to the group the route map for the location will be displayed for the users.

CHAPTER 7

CODING AND TESTING

7.1 CODING

Once the design aspect of the system is finalizes the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

7.2 CODING STANDARDS

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Program should be simple, clear and easy to understand.

Naming conventions

Value conventions

Script and comment procedure

Message box format

Exception and error handling

7.2.1 NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be self-descriptive. One should even get the

meaning and scope of the variable by its name. The conventions are adopted for easy understanding of the intended message by the user. So it is customary to follow the conventions. These conventions are as follows:

Class names

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

Member Function and Data Member name

Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in lowercase.

7.2.2 VALUE CONVENTIONS

Value conventions ensure values for variable at any point of time. This involves the following:

- Proper default values for the variables.
- Proper validation of values in the field.
- Proper documentation of flag values.

7.2.3 SCRIPT WRITING AND COMMENTING STANDARD

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

7.2.4 MESSAGE BOX FORMAT

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

- X – User has performed illegal operation.
- ! – Information to the user.

7.3 TEST PROCEDURE

SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

7.4 TEST DATA AND OUTPUT

7.4.1 UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

7.4.2 FUNCTIONAL TESTS

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test

7.4.3 PERFORMANCE TEST

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

7.4.4 STRESS TEST

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

7.4.5 STRUCTURED TEST

Structure Tests are concerned with exercising the internal logic of a program

and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

- Exercise all logical decisions on their true or false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.
- Checking attributes for their correctness.
- Handling end of file condition, I/O errors, buffer problems and textual errors in output information

7.4.6 INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all

the modules are combined the link is not set properly with all support files. Then

we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

7.5 TESTING TECHNIQUES / TESTING STRATERGIES

7.5.1 TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet – undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

7.5.1.1 WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

7.5.1.2 BLACK BOX TESTING

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

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

7.5.2 SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software

developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works "outward" toward the integration of the entire computer based system.
- Different testing techniques are appropriate at different points in time.
- The developer of the software and an independent test group conducts testing.
- Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

7.5.2.1 INTEGRATION TESTING:

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

7.5.2.2 PROGRAM TESTING:

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error

messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

7.5.2.3 SECURITY TESTING:

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

7.5.2.4 VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

* The function or performance characteristics confirm to specifications and are accepted.

* A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic

7.5.2.5 USER ACCEPTANCE TESTING

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

- Input screen design.
- Output screen design.

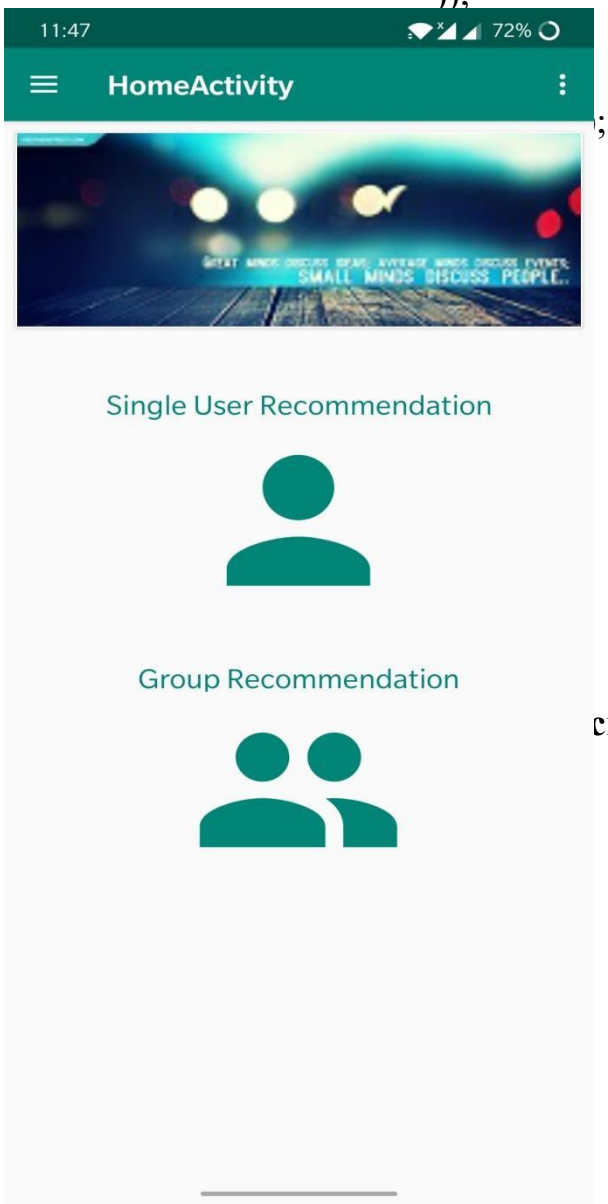
Source Code

```
<?php
require_once('DbConnect.php');
$name = $_POST['name'];
$sql = "SELECT * FROM tags";
```

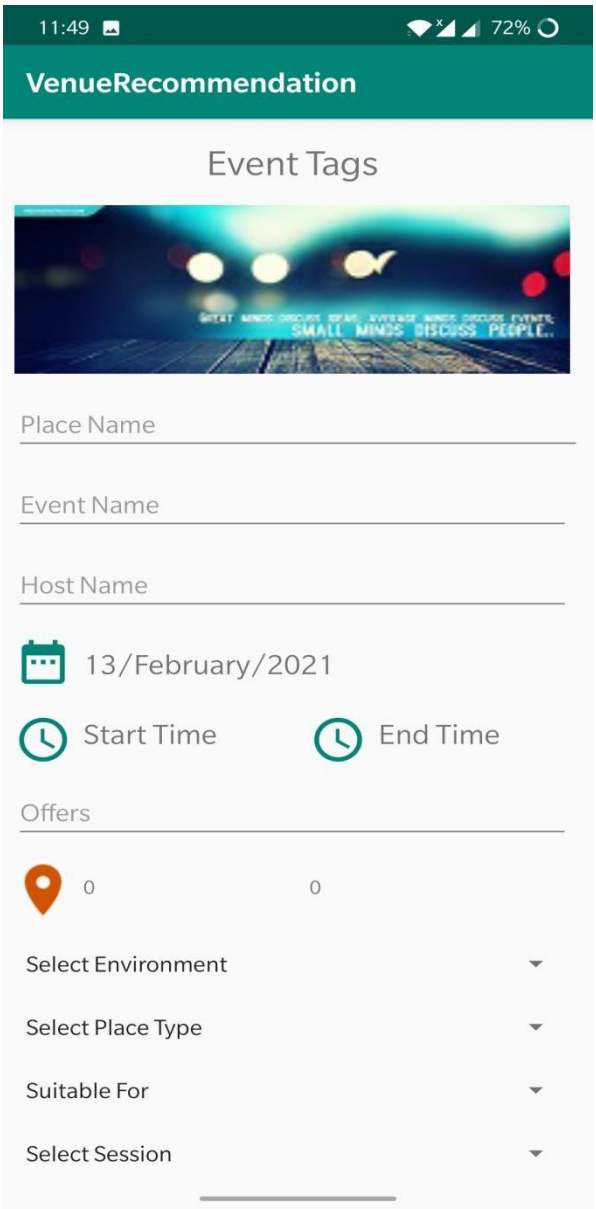
```
$r = mysqli_query($conn,$sql);
$response = array();
$result = array();
$ip = $_SERVER['SERVER_ADDR'];

while($row =
   mysqli_fetch_array($r)){ array_pus
   h($result,array(
      'name'=>$row['name'],
      'lat'=>$row['lat'],
      'lng'=>$row['lng'],
      'eventname'=>$row['eventname'],
      'hostname'=>$row['hostname'],
      'sdate'=>$row['sdate'],
      'stime'=>$row['stime'],
      'etime'=>$row['etime'],
      'offer'=>$row['offer'],
      'category'=>$row['category'],
      'placetype'=>$row['placetype'],
      'fees'=>$row['fees'],
      'description'=>$row['description'],

'eventimage'=>'http://'.$ip.'/VenueRecommender/uploadedFiles/'.$row['eventimage
'].'.jpg',
      'suitablefor'=>$row['suitablefor'],
      'session'=>$row['session']

   ));
}
$response['error'] = false;
$response['message'] = 'User registered successfully';
$response['user'] = $result;
echo json_encode($response);

mysqli_close($conn);
?>


<?php
require_once('DbConnect.php');
```

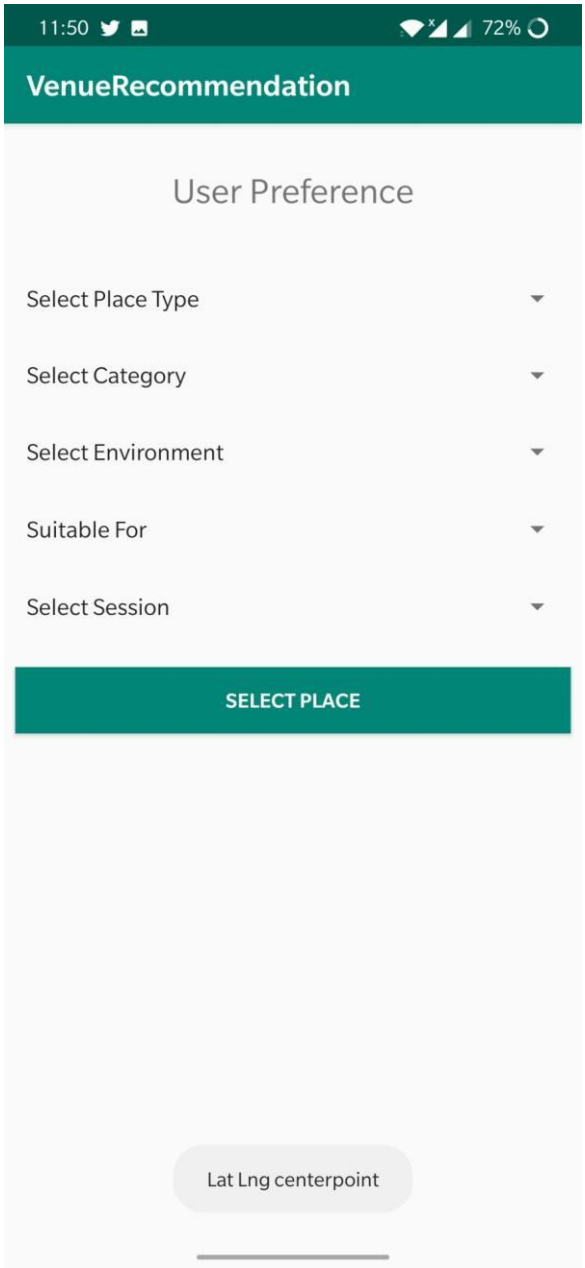
```
$uid = $_GET['uid'];


        $sql = "SELECT DISTINCT gname from cgroup where uid = '$uid'";

        $result = mysqli_query($conn,$sql);

        $res = array();

        while($row =
                mysqli_fetch_array($result)){ array_push($res,
                array(
                        "gname"=>$row['gname'],
                        ));
```

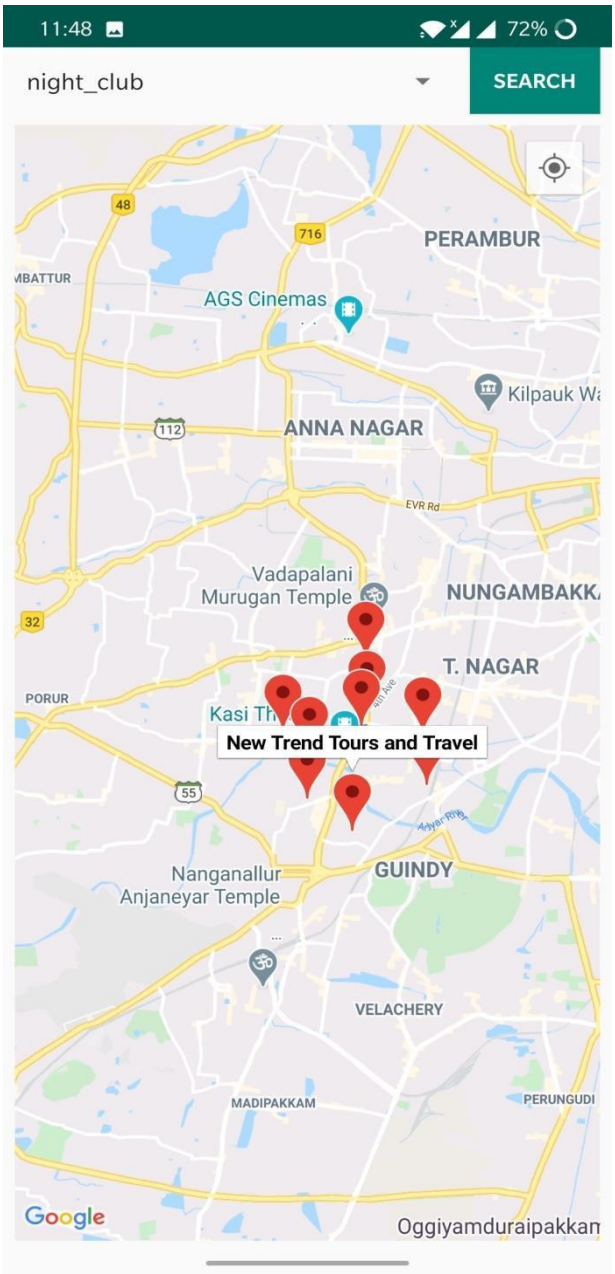
;



creenshots




11:50
72%
VenueRecommendation
User Preference
Select Place Type
Select Category
Select Environment
Suitable For
Select Session
SELECT PLACE
Lat Lng centerpoint


11:48
72%
night_club
SEARCH
PERAMBUR
AGS Cinemas
Kilpauk Wa
ANNA NAGAR
EVR Rd
Vadapalani
Murugan Temple
NUNGAMBAKK
T. NAGAR
PORUR
New Trend Tours and Travel
Nanganallur
Anjaneyar Temple
GUINDY
VELACHERY
MADIPAKKAM
PERUNGUDI
Google
Oggiyamduraipakkam

REFERENCES

[1] A. Q. Macedo, L. B. Marinho, and R. L. Santos, “Context-aware event

recommendation in event-based social networks," in Proceedings of RecSys '15. New York, NY, USA: ACM, 2015, pp. 123– 130.

[2] Z. Qiao, P. Zhang, Y. Cao, C. Zhou, L. Guo, and B. Fang, "Combining heterogenous social and geographical information for event recommendation," in Proceedings of AAAI Conference on Artificial Intelligence, July 27 -31., 2014, pp. 145–151.

[3] Q. Yuan, G. Cong, and C.-Y. Lin, "COM: A Generative Model for Group Recommendation," in Proceedings of KDD '14. New York, NY, USA: ACM, 2014, pp. 163–172.

[4] W. Zhang, J. Wang, and W. Feng, "Combining latent factor model with location features for event-based group recommendation," in Proceedings of KDD '13. New York, NY, USA: ACM, 2013, pp. 910–918.

[5] X. Liu, Q. He, Y. Tian, W.-C. Lee, J. McPherson, and J. Han, "Event-based social networks: Linking the online and offline social worlds," in Proceedings of KDD '12. New York, NY, USA: ACM, 2012, pp. 1032–1040.

[6] S. Purushotham and C.-C. J. Kuo, "Personalized group recommender systems for location- and event-based social networks," ACM Trans. Spatial Algorithms Syst., vol. 2, no. 4, pp. 16:1–16:29, Nov. 2016.

[7] P. Lops, M. De Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in Recommender handbook. Springer, 2011,

pp. 73–105.

[8] A. Mnih and R. R. Salakhutdinov, “Probabilistic matrix factorization,” in Procedings of NIPS ’08, 2008, pp. 1257–1264.

[9] B. Liu, H. Xiong, S. Papadimitriou, Y. Fu, and Z. Yao, “A general geographical probabilistic factor model for point of interest recommendation,” IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 5, pp. 1167–1179, May 2015.

[10] H. Yin, X. Zhou, B. Cui, H. Wang, K. Zheng, and Q. V. Nguyen, “Adapting to user interest drift for poi recommendation,” IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 10, pp. 2566–2581, Oct. 2016.