1. **INTRODUCTION:-**

The Internet has revolutionized the way we connect and communicate with people. It helps connect with people far away and informs of everything that is going on in the world.

There are various events, happenings that take place near where we work, or stay or visit frequently. We are oblivious to most of the events that happen in our surroundings as there are few applications that use the internet to notify the about such events. Vicinity explorer takes advantages of networks to explore this crucial area.

The system helps users discover the area that’s most important to him/her (Home, Workplace or any other active location). It creates a geological bound within the users’ location and provides a set of features to the user pertaining to that location. The app is bound to a geographic location and prioritizes depth of content over breath.

1. Local offers, openings and other important events with PORTALS.
2. Sharing books, DVDs and other materials.
3. Discovering places and reviewing them locally.
   1. **PURPOSE OF THE PROJECT:-**

The purpose of the project is to allow users to take advantage of the Internet to explore their surroundings. The system will provide an easy-to-access infrastructure, which will help users to track all the events occurring near their home location in real time. As the location is fixed, there is no need for a GPS enabled smart phone. With push notification services, users will always be notified about all the happenings around their location and thus, they will never miss any important event.

This system also provides a great framework for location based advertisement. It allows content to be published locally for greater impact on customers. It is a boost for those who want to restrain their advertising area.

* 1. **SCOPE OF THE PROJECT:-**

The scope of the system can be described by the services which it provides. The core service the application provides is called as “Portal”. This allows anyone to publish content locally. This is a great way to advertise events, sales etc. The Portal Mechanism is much more general and can be used for geo-multicasting news about events, offers, community formations as well as awareness system for any natural disaster. It can be used by anyone to multicast important messages pertaining to that area.

Another important service that the application provides is called local book sharing. This forms a sharing infrastructure for books or other tangible things. People nearby can trade books, DVDs, comics etc. with the help of the application.

* 1. **CONSTRAINTS OF THE PROJECT:-**

1. Internet Access.
2. All information will be location bound.
3. Restrictions on content to be published.
4. Location Constraints.
   1. **ASSUMPTIONS AND DEPENDENCIES:–**
5. All the locations are assumed to be discrete blocks.
6. Users should have GPRS connection in their phone.
7. **LITERATURE REVIEW:-**

**EXISTING SYSTEM:-**

The college experience is multi-faceted, but not enough has been done in the past to extract personal informatics from daily student life. OmniStanford [1], a smartphone application that automatically logs on-campus location history. Using this information, we are able to introduce two new experience-enhancing features: discovery and journaling. The discovery feature allows students to find people with common characteristics who were at the same location at the same time. The journaling feature gives students the opportunity to associate tasks to locations, enabling personal tracking and recall.

**RELATED WORK:-**

Musubi is a disintermediated messaging application for Android devices. Users can join with any of their existing identities (e.g. Facebook, Google, and Stanford) and share words, pictures, and videos with friends. Developers can build applications such as multiplayer games on top of Musubi. All Musubi messages are encrypted using Identity Based Encryption, and no central server stores any message. Personal informatics has been an active area of research in recent years. For example, Aseniero, et al. discuss visualizing daily activities, though not necessarily targeted to students, and focuses on viewing data on a macro level. Our journaling tool examines activities at multiple levels of granularity, though this work could serve as a supplement to ours. [1]

**PEOPLE DISCOVERY:-**

The first application we present for location-based PI is discovering individuals with common personal attributes, such as department affiliation or residence. We present an interface through which a user can select his affiliations, along with an option to allow location sharing with other application users. OmniStanford automatically connects to Musubi to allow communication through existing identities with a common format. We provide a server component for each location that can look up the names and hashed identifiers of people who have matching characteristics. For example, if two new Computer Science students meet at orientation, their presence at the same event will be pushed to each student by the server due to a match on the Computer.

We provide a daily calendar-like interface to list all the check-ins for a day and a drag-and-drop interface for the user to assign a tag to a check-in. The tag is created by user and can represent activities, food, physical exercises, and others. The application will remember the most recent tags so users do not need to type them every time. Once tags are assigned to check-ins, user can choose to visualize them invarious ways. Currently we only built one view in D3.js: a pie chart that visualizes user’s daily, weekly, or monthly data. Both the selection interface and the D3 visualization are shown in Figure 1. The pie chart displays the portion of time spent at each location and the portion of time for each activity, which is personally meaningful for user. Aseniero [1] et al. built similar PI tools and we found that our two level pie chart meets most users’ needs. Besides reflection, journaling can also help students manage their time and change their behavior in the future.

**MULTICASTING ALGORITHM:-**

Flooding is probably the simplest multicast routing algorithm.The flooding algorithm can be used to deliver packets to nodes within a location-based multicast group. The multicast flooding algorithm can be implemented as follows: Assume that a node S needs to send a packet to a specific multicast region, a circle in figure. Node S broadcasts the multicast packet to all its neighbors\_ – hereafter, node S will be referred to as the *sender* and nodes D, F, and G as the *multicast group members* (note that in Figure 1 all nodes present in the specified multicast region are, by definition, multicast group members). A node, say B or C, on receiving the packet, compares the specified region’s coordinates with its own location. (We assume that all hosts are able to determine their own location using GPS.) If the location of B is within the specified multicast region, node B will accept the packet. Node B will also broadcast the packet to its neighbors; if it has not received the packet previously (repeated reception of a packet is detected using sequence numbers). If node B is located outside the multicast region and the packet was not received previously, it just broadcasts the packet to its neighbors.[4]

Yes

No

No

Start

Yes

Redundant nodes exist?

End

*Tk*🡨*Ttrim*

*Tk*🡨*TMST*

*e*(*Tk*)>*e*(*TMST*)?

*TMST*: use the nodes in *Tk* to generate an MST

*Ttrim*: delete the redundant nodes and edges

*Tk*: a tree built by an ant *k*

Fig. 2.1 Flowchart for Multicasting Algorithm

1. **OVERALL SYSTEM DESCRIPTION:-**
   1. **EXISTING SYSTEM:-**

The first application present for location-based PI is discovering individuals with common personal attributes, such as department affiliation or residence. We present an interface through which a user can select his affiliations, along with an option to allow location sharing with other application users. OmniStanford automatically connects to Musubi to allow communication through existing identities with a common format. We provide a server component for each location that can look up the names and hashed identifiers of people who have matching characteristics. For example, if two new Computer Science students meet at orientation, their presence at the same event will be pushed to each student by the server due to a match on the Computer. We provide a daily calendar-like interface to list all the check-ins for a day and a drag-and-drop interface for the user to assign a tag to a check-in. The tag is created by user and can represent activities, food, physical exercises, and others. The application will remember the most recent tags so users do not need to type them every time. Once tags are assigned to check-ins, user can choose to visualize them in various ways. The pie chart displays the portion of time spent at each location and the portion of time for each activity, which is personally meaningful for user. Ascenario et al built similar PI tools and we found that our two level pie chart meets most users’ needs. Besides reflection, journaling can also help students manage their time and change their behavior in the future.

* 1. **PROPOSED SYSTEM:-**

The App (Web, Mobile and Desktop) will require the user to input his HOME location, with this location it can give the following services to the user Location Based Friend Search, Food Zones, Help, Marketing Portals, Books and CDs sharing, Location Based News and Offers Roller, Campaign Launcher, Group Formation and Meet Ups, Startups, Jobs and Internships, Real-time Location helps.

The application opens a new world that lies in your geographic bound, a small world that is yet to be explored, studied and experienced.The application is bound to your geographic bounds and prioritizes depth of content over breath.

We use Machine learning to constantly know your behavior and help you find people, places, jobs that are closest to your location which you may not know.

It will give you real-time updates about the happenings around you in the News Roller.

It will help marketing and consumer communication as organizations can easily advertise their products locally according to their needs and can create special zones relevant to their products.

It will feature Issue Clustering where users can write in their queries or any help required for any task, any nearby user can help them and earn points when the job is done. As users are nearby the chance of getting help increases rather than a worldwide or nationwide service.

Food zone will help you find need restaurants and diners near your location and will allow you to give feedback and rating which can be helpful for the owners as well as to others living near you. Again, the application prioritizes location and only shows you the places near you.

Book sharing allows users to share books by creating a virtual library where users can trade books and save money.

Jobs, startups and internships allow the firms and startups to locate nearby employees who satisfy their needs. Nearby jobs save travelling and help both the firm and the employee. Startups will get some benefits by getting local personnel.

Campaign launcher is a special feature which allows NGO and other organization to gather local crowd and start some campaigns for some good cause. This will allow youngsters to be a part of the cause. The communication can be consolidated by the means of the application.

Sometimes when we aren't at home and still want some services such as nearby movies, café or emergency service the app can still provide you with real-time service by obtaining your current location.

* 1. **METHODOLOGY:-**

**Process Model: Waterfall Model:-**

The approach that is used to develop the project will be object oriented approach. It is intended to develop the system components as objects and interconnect the system components by message passing between the objects. The implementation of various features of the project involves generation of different classes.

The process model to be used for this project will be the waterfall model as shown in the figure. Using this method, the features of the project will be developed by following the steps of the system development life cycle linearly i.e. requirement gathering, analysis, design, implementation, testing and deployment. For any changes to the system, the same procedure will be followed, but with the use of agility model. In this project the steps of system development life cycle may be skipped or speeded up so that the component design is generated faster to meet the deadlines.

Deployment

Testing

Analysis

Implementation

Design

Requirement gathering Gathering

Fig.3.1 Phases of Waterfall Model

1. **REQUIREMENTS GATHERING AND PLANNING:-**
   1. **REQUIREMENTS ELICITATION :-**
      1. **USE CASE DIAGRAM:-**



Fig. 4.1 Use Case

* + 1. **CLASS DIAGRAM:-**



Fig. 4.2 Class Diagram

* 1. **FEASIBILITY STUDY:-**
     1. **Technical Feasibility:-**
        1. **Hardware Requirement:-**
* Application Server : Apache 2.4
* Database Server: MySQL, MongoDB
* Internet Connection (At least 512Kbps)
  + - 1. **Software Requirement:-**
* Web Browser: Mozilla Firefox 20+, Google Chrome 25+, Internet Explorer 9+
* My SQL Server
* J2ME
  + 1. **Economic Feasibility:-**
* Server System: Rs. 10,000/yr.
  1. **REQUIREMENTS ANALYSIS:-**
     1. **FLOW CHART:-**

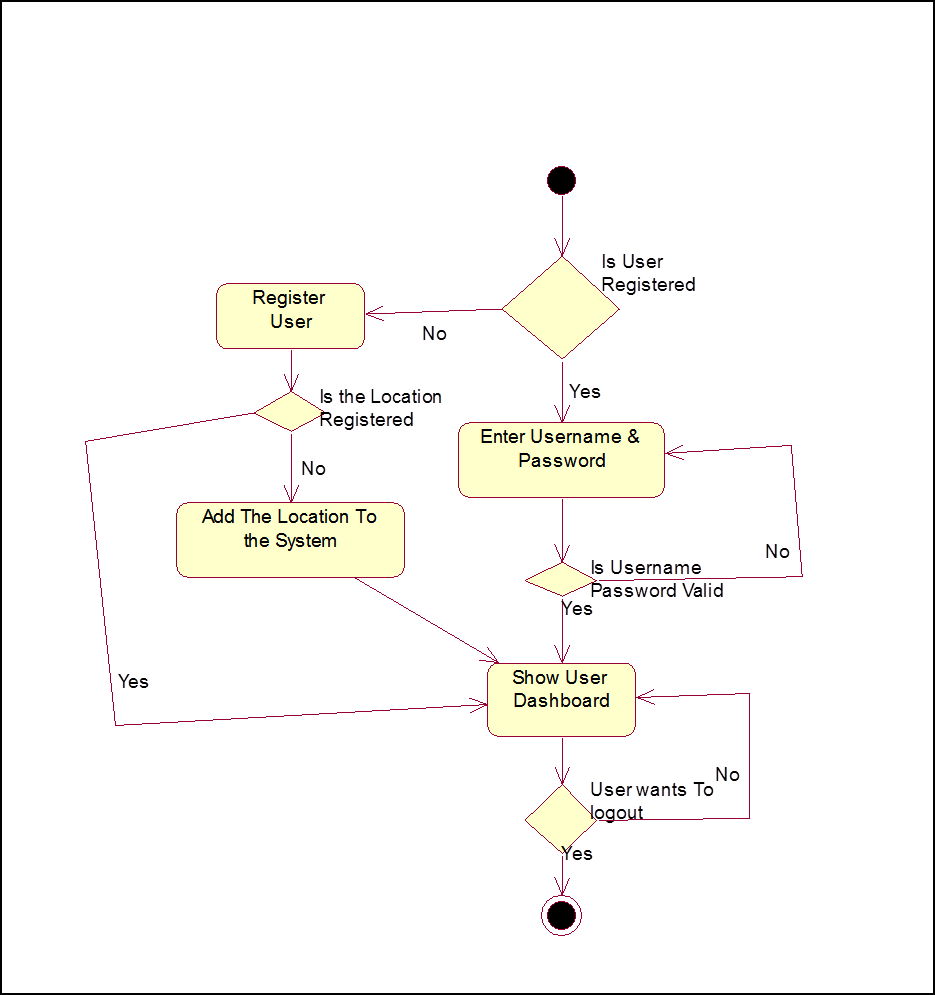


Fig. 4.3 Validating User



Fig. 4.4 Adding Content to the Dashboard

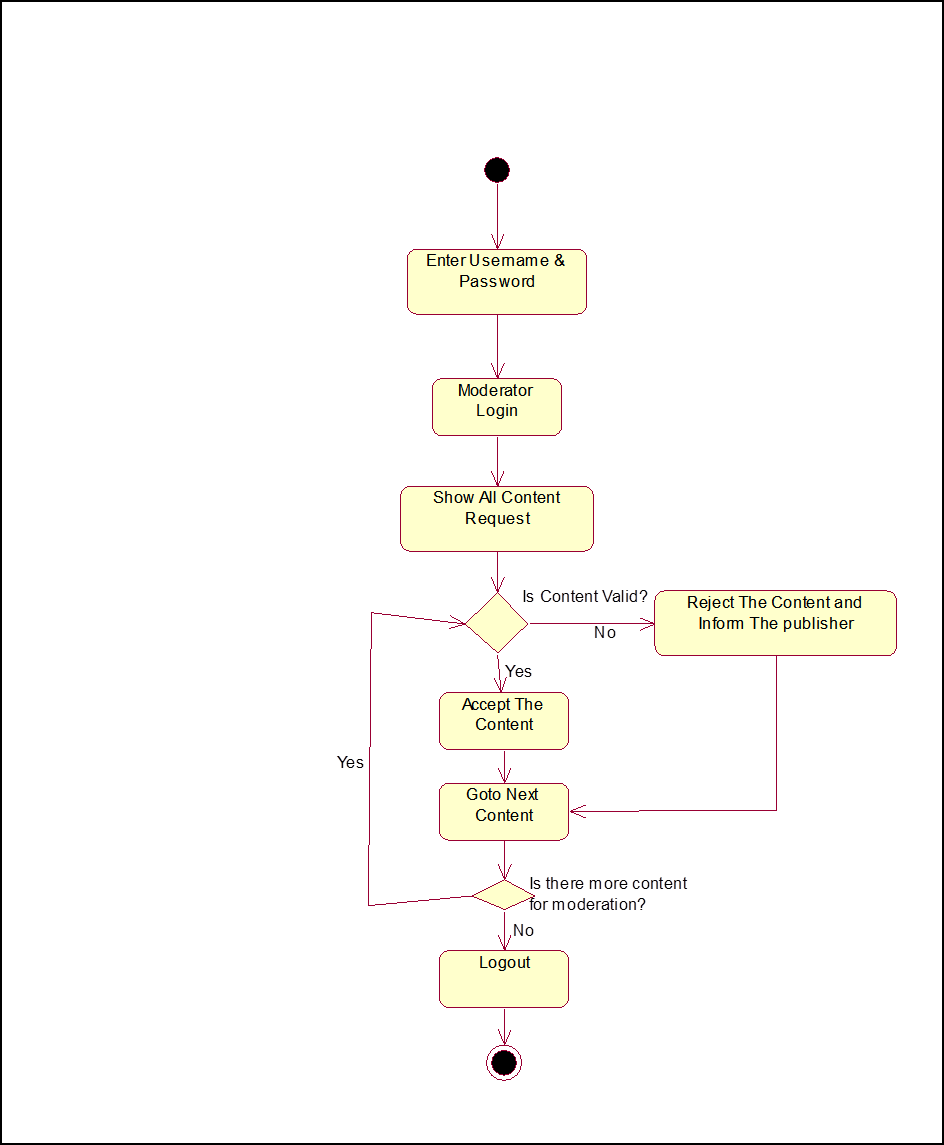


Fig. 4.5 Validating the Content

* 1. **TIME LINE CHARTS:-**

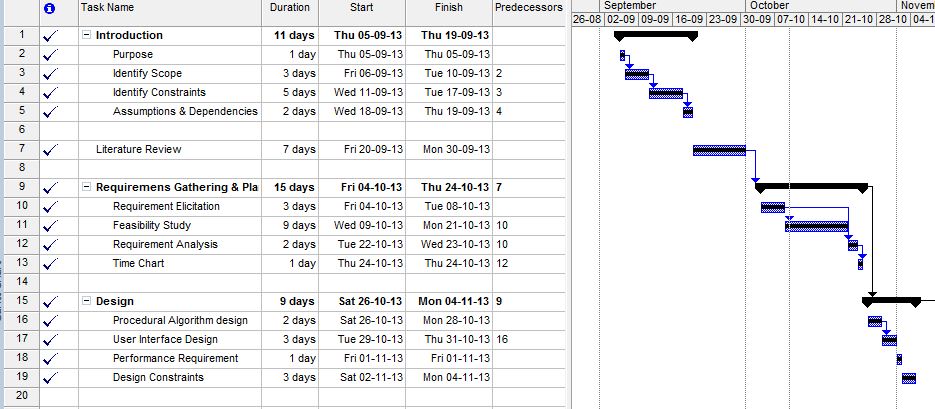


Fig. 4.6 Time Line Chart 1

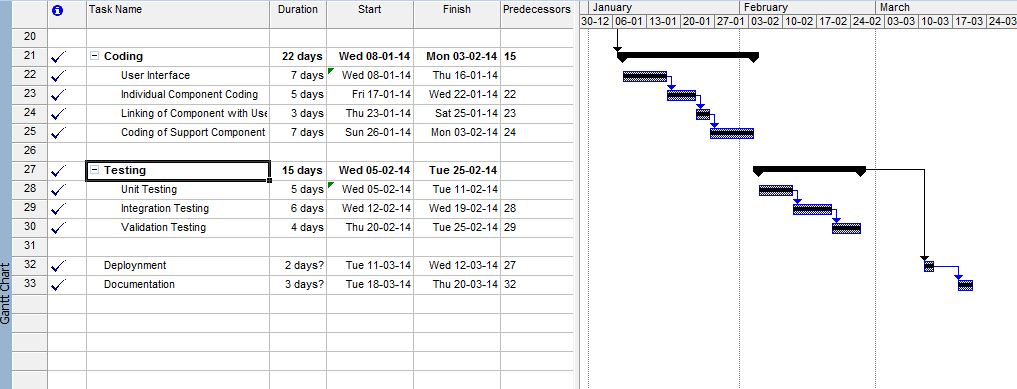


Fig. 4.7 Time Line Chart 2

1. **ANALYSIS:-**
   1. **. ENTITIES AND RELATION:-**
      1. **List of Entities**
2. User
3. Publisher
4. Moderator
5. Book
6. Post
7. Location

**5.1.2. List of Relations**

1) User to location

2) User to Books

3)Location to Content

4) Content to Publisher

5) Location to Moderator

* 1. **ERD:-**

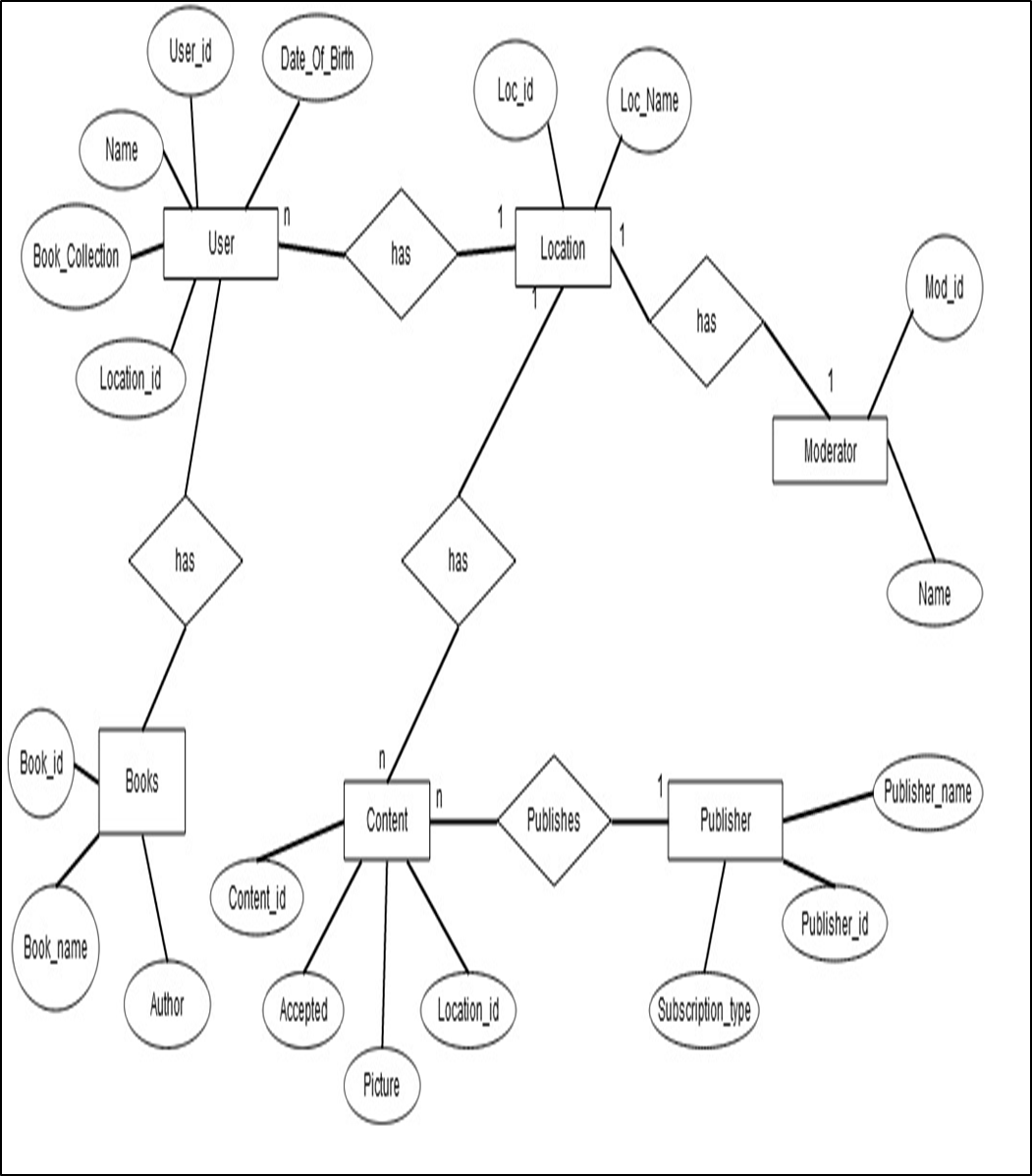
**

Fig. 5.1 Entity Relationship Diagram

**5.3 DATA DICTIONARY:-**

User

|  |  |  |  |
| --- | --- | --- | --- |
| User\_Id | Number(10) | Primary Key | Unique Identification Number |
| Name | Varchar(20) | NOT NULL | User’s Full Name |
| Location\_Id | Number(10) | NOT NULL |  |
| Book\_Collection | Array |  | Collections of books user is sharing |
| Date Of Birth | Date | NOT NULL | User’s DOB |
| Contact Number | Number(10) |  | Users Contact Number |

Publisher

|  |  |  |  |
| --- | --- | --- | --- |
| Publisher\_Id | Number(10) | PRIMARY KEY | Unique Publisher Identification |
| Subscription\_type | Varchar(10) | NOT NULL | Monthly or Yearly |
| Publisher\_Name | Varchar(40) | NOT NULL | Name of the Firm or Individual |

Book

|  |  |  |  |
| --- | --- | --- | --- |
| Book\_id | Number(10) | PRIMARY KEY | Unique Book Identification |
| Book\_name | Varchar(20) | NOT NULL | Name of the Book |
| Author | Varchar(20) | NOT NULL | Author’s Name |
| Date\_of\_sharing | Date | NOT NULL | When the book was shared |

Moderator

|  |  |  |  |
| --- | --- | --- | --- |
| Moderator\_id | Number(10) | PRIMARY KEY | Unique Moderator Identification |
| Moderator\_name | Varchar(30) | NOT NULL | Name of Moderator |

Post

|  |  |  |  |
| --- | --- | --- | --- |
| Post\_Id | Number(10) | PRIMARY KEY | Unique Moderator Identification |
| Content | Varchar(400) | NOT NULL | Text part of content |
| Image | File |  | Image of the content |
| Moderator\_id | Foreign Key | NOT NULL | Moderators id. |
| Publisher\_id | Foreign Key | NOT NULL | The Id of the publisher |

**5.4 SEQUENCE DIAGRAM:-**



Fig. 5.2 Sequence Diagram

1. **DESIGN:-**
   1. **WORK BREAKDOWN STRUCTURE:-**

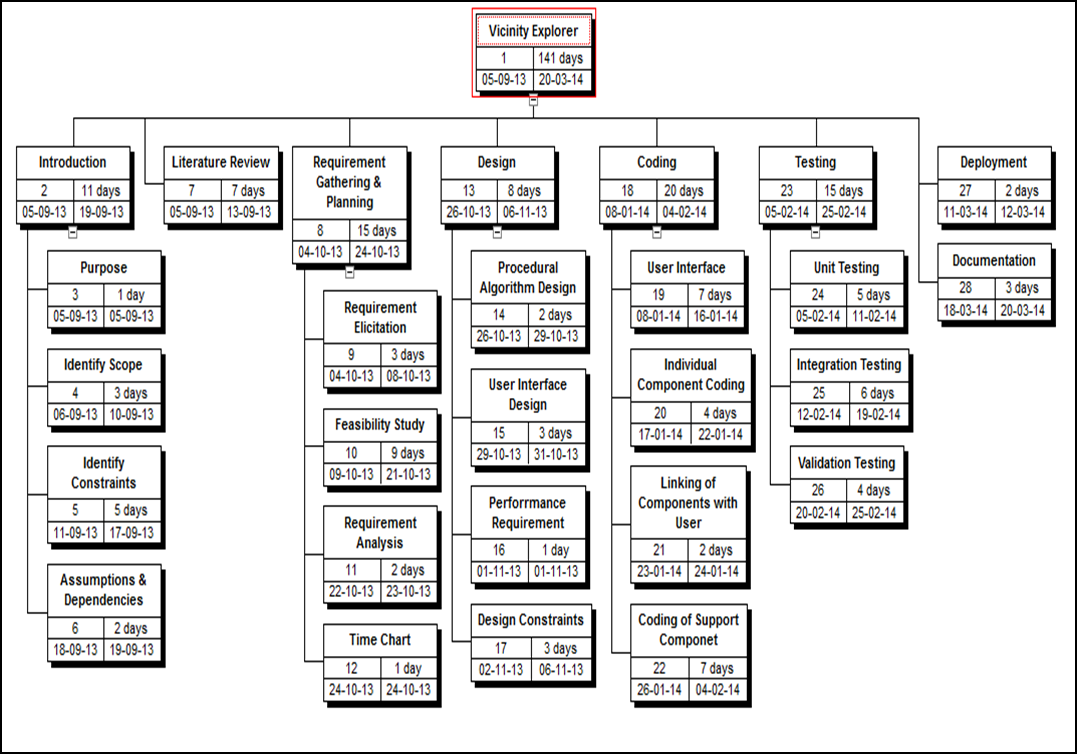


Fig. 6.1 Work Breakdown Structure

**6.2. PROCEDURAL DESIGN:-**

|  |  |
| --- | --- |
| Process Name | Register |
| Input | Username  Password  Location Name(Home Location) |
| Output | Confirmation Mail |
| Description | This is a user side process  This Process is used to register the  user. |

|  |  |
| --- | --- |
| Process Name | Get Contents |
| Input | Username  Location Name |
| Output | Contents For that location |
| Description | This provides the user with all the  contents relevant to the location  name |

|  |  |
| --- | --- |
| Process Name | Share Book |
| Input | Book Name  User Name  Location Name |
| Output | Confirmation of sharing |
| Description | This allows the users to share books |

|  |  |
| --- | --- |
| Process Name | Publish Contents |
| Input | Publisher Name  Location Name |
| Output | Request Verification |
| Description | This allows publisher to publish  contents to that location |

|  |  |
| --- | --- |
| Process Name | Validate Content |
| Input | Content  Location Name  Publisher Name |
| Output | Content Validation Message – Valid  Or  Invalid |
| Description | The validation of the content is  performed by the moderator. |

**6.3.UI DESIGN (SNAPSHOTS)/LAYOUTS:-**

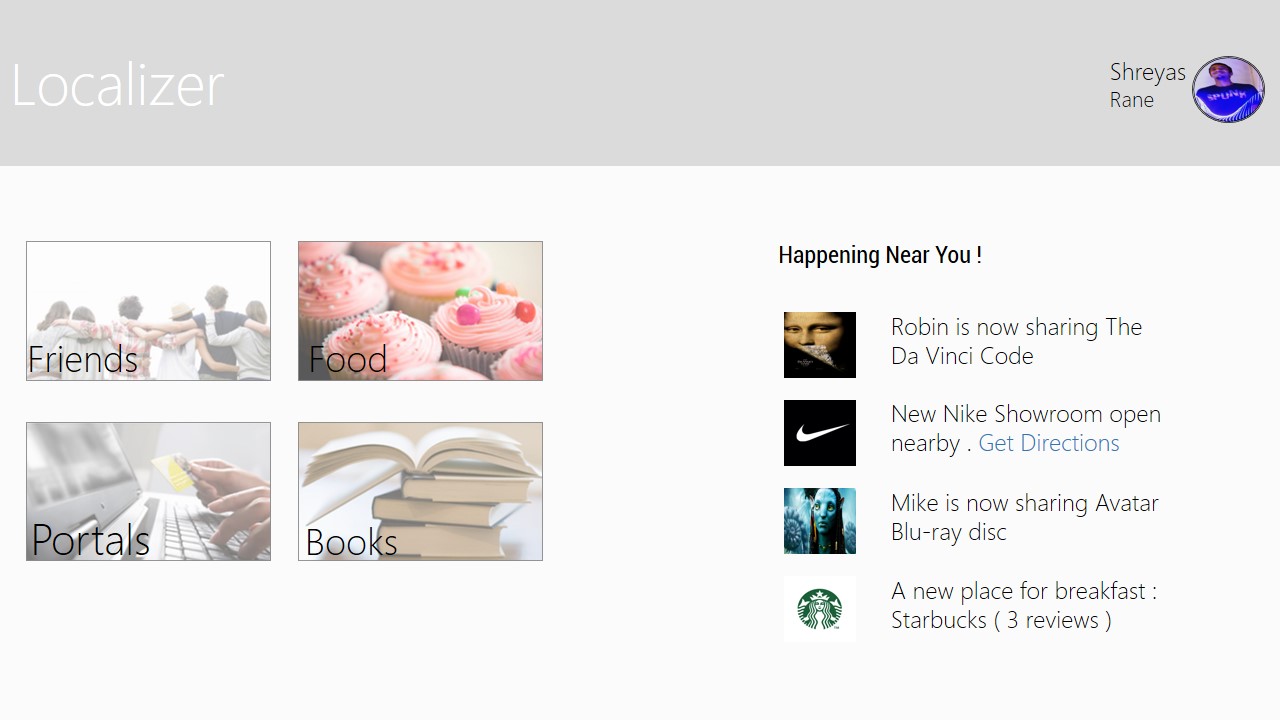
****

Fig. 6.2 User Interface of the Site

**6.4.PERFORMANCE REQUIREMENT:-**

* + 1. Capacity Requirements
* No of Users : 10 lakhs
* Max no of Transactions : 10 lakhs
* No of tables in Database : 6
  + 1. Response Time
* Response time will be depends on the server load.
* Speed of processor: 5GHz (8 core)
  + 1. Throughput
* Throughput in seconds: 120.67sec/10000request
  1. **DATA FLOW DIAGRAMS:**

content

location

USER

PUBLISHER

content

ACK

Fig. 6.3 Level 0 DFD for Vicinity Explorer System

content

content

PUBLISHER

ACK

content

content

USER

location

Fig. 6.4 Level 1 DFD for Vicinity Explorer System

location

location

USER

content

Fig. 6.5 Level 2 DFD for Geo Multicasting

1. **APPENDIX:-**

MULTICASTING ALGORITHM:

Flooding is probably the simplest multicast routing algorithm .The flooding algorithm can be used to deliver packets to nodes within a location-based multicast group. The multicast flooding algorithm can be implemented as follows: Assume that a node S needs to send a packet to a specific multicast region, a circle in figure. Node S broadcasts the multicast packet to all its neighbors\_ – hereafter, node S will be referred to as the *sender* and nodes D, F, and G as the *multicast group members* (note that in Figure 1 all nodes present in the specified multicast region are, by definition, multicast group members). A node, say B or C, on receiving the packet, compares the specified region’scoordinates with its own location. (We assume that all hosts are able to determine their own location using GPS.) If the location of B is within the specified multicast region, node B will accept the packet. Node B will also broadcast the packet to its neighbors; if it has not received the packet previously (repeated reception of a packet is detected using sequence numbers). If node B is located outside the multicast region and the packet was not received previously, it just broadcasts the packet to its neighbors.

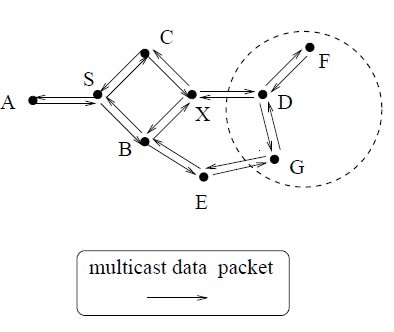


Fig. 7.1 Multicasting Packets

1. **BIBLIOGRAPHY AND REFERENCES:-**
2. Cain Evans and Philip Moore and Andrew Thomas, “An Intelligent Mobile Advertising System (iMAS) Location-Based Advertising to Individuals and Business”. IEEE 2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems
3. KanakBiscuitwala and Steve Fan “OmniStanford: Location-Based

Discovery and Journaling for College Students” UIST’12, October 7-10, 2012, Cambridge, MA, USA.

1. Dodson, B., Vo, I., Purtell, T., Lam, M.S. “Musubi: Disintermediated Interactive Social Feeds for Mobile Devices”. In Proceedings of the 21st International Conference on World Wide Web. ACM, 2012.
2. Young-BaeKo and Nitin H. Vaidya ”Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms”

1. I.F.Akyildiz, S. M. Joseph, and Y.-B. Lin. “Movement based location update and selective paging for PCS networks”. IEEE/ACM Trans. on Networking, 4(4):94–104, 1996.
2. Aseniero, B.A., Carpendale, S., Tang, A. “Deep Personalization in Tools for Reflection” In CHI 2012 Workshop on Personal Informatics in Practice: Improving Quality of Life through Data, Austin, TX.