CLOUD PROJECT REPORT

RePOST! – A web based service

Arvind Nair & Pulkit Sood

DEPARTMENT OF COMPUTER & INFORMATION SCIENCE, IUPUI

Contents

1. Introduction 5
2. Basic Functionalities 5
Flow of the Website 6
3. Website Model 7
Use Case Model 7
1. General View 7
2. Use Case View 8
4. Website Pages 9
1. Login Page 9
2. Registration Page 9
Login and Registration Page Sequence 10
3. Home Page 10
Home Page Review Sequence 11
4. Follow Page 11
Follow Sequence 12
5. Chat Page 12
Chat Sequence 13
6. GMap Page 14
GMap Sequence 15
5. Database Model of the system 16
6. Additional Features of the Website 17
Posting a Review Features 17
Vote Page 17
Vote Statistics Page 17
Report Spam Page 17
Spam Statistics Page 18
7. Deployment on the cloud 18
Part 1: Creating an instance/virtual machine on the cloud 18
Part 2: Installing the required software on the virtual instance 18
Step 3: Transferring files to the cloud 19
Step 4: Database deployment on the cloud 19

8. Benefits of Deployment on Cloud 19
1. Minimal Cost 19
2. Focus on Website 19
3. Speed of Deployment 20
4. Ease of Use and Management 20
5. High Scalability 20
6. Better Business Model 20
7. Redundancy and Resilience 20
8. Legal Contract 20
9. Technologies Applied 20
1. Technologies 20
2. Development Environment 20
10. Screen Shots 21
11. Future Work 46
12. Contribution by Team Members 47
Contribution by Arvind Nair 47
Contribution by Pulkit Sood 47
Contribution done as a team 47
13. References 47
14. Appendix 47

Table of Figures

Figure 1 : Flow Diagram 6
Figure 2 : General View 7
Figure 3 : Use Case View 8
Figure 4 : Login & Registration Sequence Diagram 10
Figure 5 : Home Page Review Sequence Diagram 11
Figure 6 : Follow Sequence Diagram 12
Figure 7 : Chat Sequence Diagram 13
Figure 8 : GMap Sequence Diagram 15
Figure 9 : Diagrammatic representation of the database using ERWin® 16
Figure 10 : Registration Page for PC 21
Figure 11: Registration page on Android 21
Figure 12: First post for PC User 22
Figure 13: First post for Android user 22
Figure 14: Follow page on PC 23
Figure 15: Follow page for Android 23
Figure 16: Wall after follow on PC 24
Figure 17 Wall after follow on Android 24
Figure 18: Search hashtag on PC 25
Figure 20: Hashtag Search on Android 25
Figure 19: Hashtag Search on Android 25
Figure 21: Voting on a review on PC 26
Figure 22: Voting on a review on Android 26
Figure 23: Confirmation message after vote on PC 27
Figure 24: Confirmation message after vote on Android 27
Figure 25: Vote stats for a review on PC 28
Figure 26: Vote stats for a review on Android 28
Figure 27: Error message for self-vote on PC 29
Figure 28: Error message for self-vote on Android 29
Figure 29: User Arvind has posted a spam message 30
Figure 30: User pulley06 has reported this as spam 30
Figure 31: User pulley06 posts a spam review 31
Figure 32: User Arvind has reported this post as spam 31
Figure 33: User Arvind views spam count on PC 32
Figure 34: User pulley06 views spam count on Android 32
Figure 35: User Arvind tries to report review as spam again 33
Figure 36: User tries to re-report spam 33
Figure 37: Error message when reporting own review as spam 34
Figure 38: Error message when reporting own review as spam 34
Figure 39: Review geo-location feature 35
Figure 40: Review geo-location feature 35
Figure 41: Prompt to add unique chat name 36

Figure 42: Prompt to add unique chat name 36
Figure 43: Blank chat view 37
Figure 44: Blank chat view 37
Figure 45: Chat image 1 38
Figure 46: Chat image 2 38
Figure 47: Chat image 3 39
Figure 48: Chat image 4 39
Figure 49: GMap on PC 40
Figure 50: GMap on Android 40
Figure 51: Arvind searches for Starbucks in Indianapolis with distance 1000 41
Figure 52: pulley06 searches for shopping places with distance 500 41
Figure 53: Same search with a larger distance 42
Figure 54: Same search with a larger distance 42
Figure 55: Error message when you delete a review not posted by you 43
Figure 56: Same error message as Figure 55 43
Figure 57: Deleted review successfully 44
Figure 58: Successfully deleted review 44
Figure 59: Message displayed after successfully unfollowing a user 45
Figure 60: Successfully unfollowed a user 45

1. Introduction

This project is a social networking site where users can create their own profiles and update their personal information, post messages which will appear on their wall/ and use a variety of features. The main purpose of website is to find out places of interest which the users will post as messages and people will up vote posts which they like. Thus, it is a review based social networking website which can be used by various enthusiasts and as an information medium for others to view.

2. Basic Functionalities

The user first registers into the website by entering the required credentials. After registration, the user logs in using her/his username and password. She/he is taken to her/his own page where she/he can view her/his own posts. There is a follow button which takes the user to the page where she/he can see other users. She/he can select a user of her/his choice to follow. After this action, the posts by the users that she/he has followed will appear on her/his wall. She/he can vote these posts, view the vote statistics for that particular post, report the post as spam and view the spam statistics for that post. The chat feature on the user's homepage will take the user to another page which is a common chat room where any users can interact with one another. The user can choose a particular name to be anonymous for security reasons. She/he can anonymously chat with other users about various topics of interest. The user on her/his home page has another feature called gmap which she/he can use to search places of interest. The user on clicking this button will be taken to a page which contains Google map modified with search options for particular places of interest. She/he can find out places of her/his choice and then visit them and post a review/information about that place to her/his wall. Other users who will be following the user can view her/his post and then vote accordingly. The user also has on her/his home page a search bar in which she/he can search posts by using hashtags (#). This will help the user to identify the posts relevant to her/his interests.

Flow of the Website

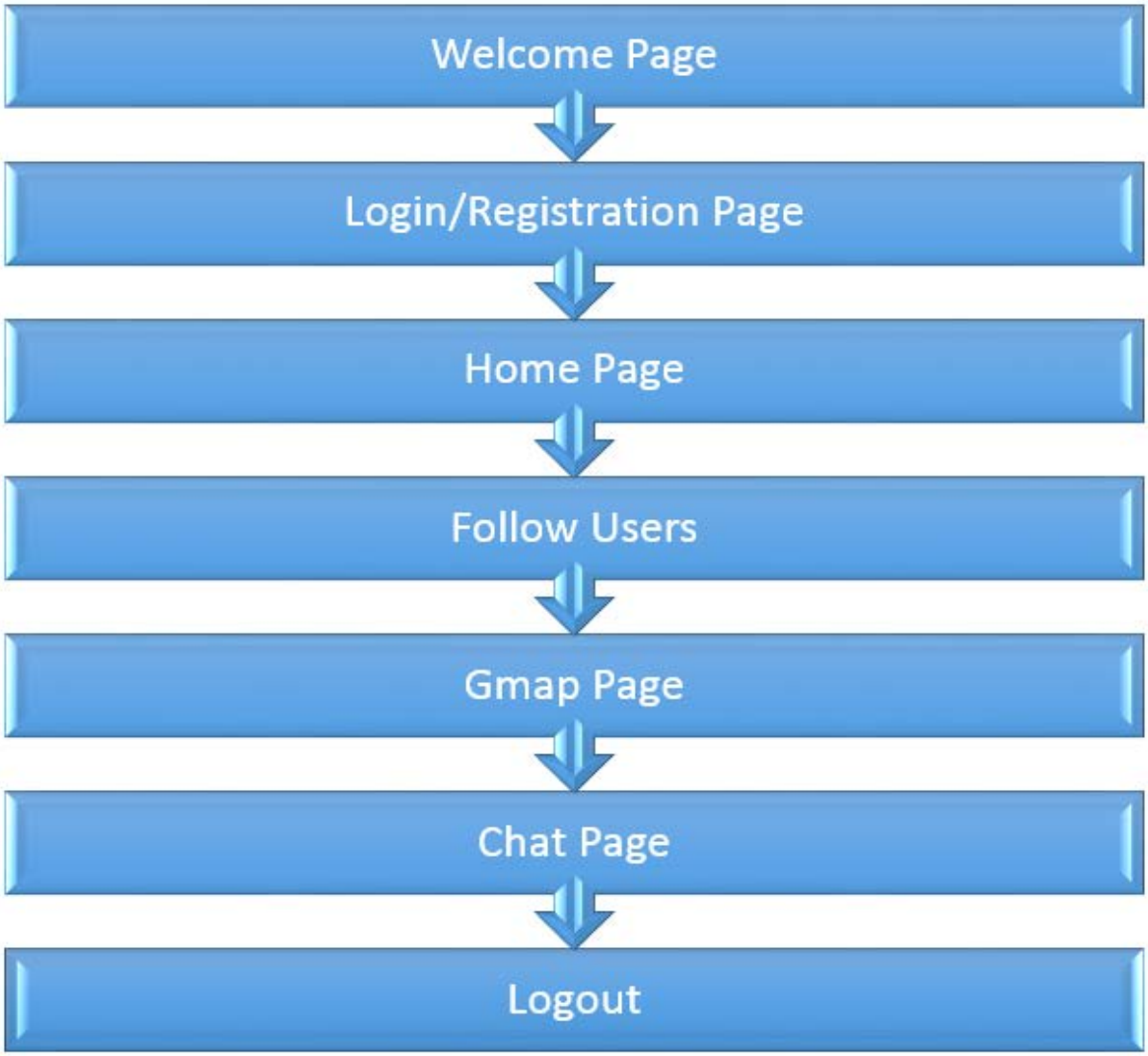


Figure 1 : Flow Diagram

3. Website Model

Use Case Model

1. General View

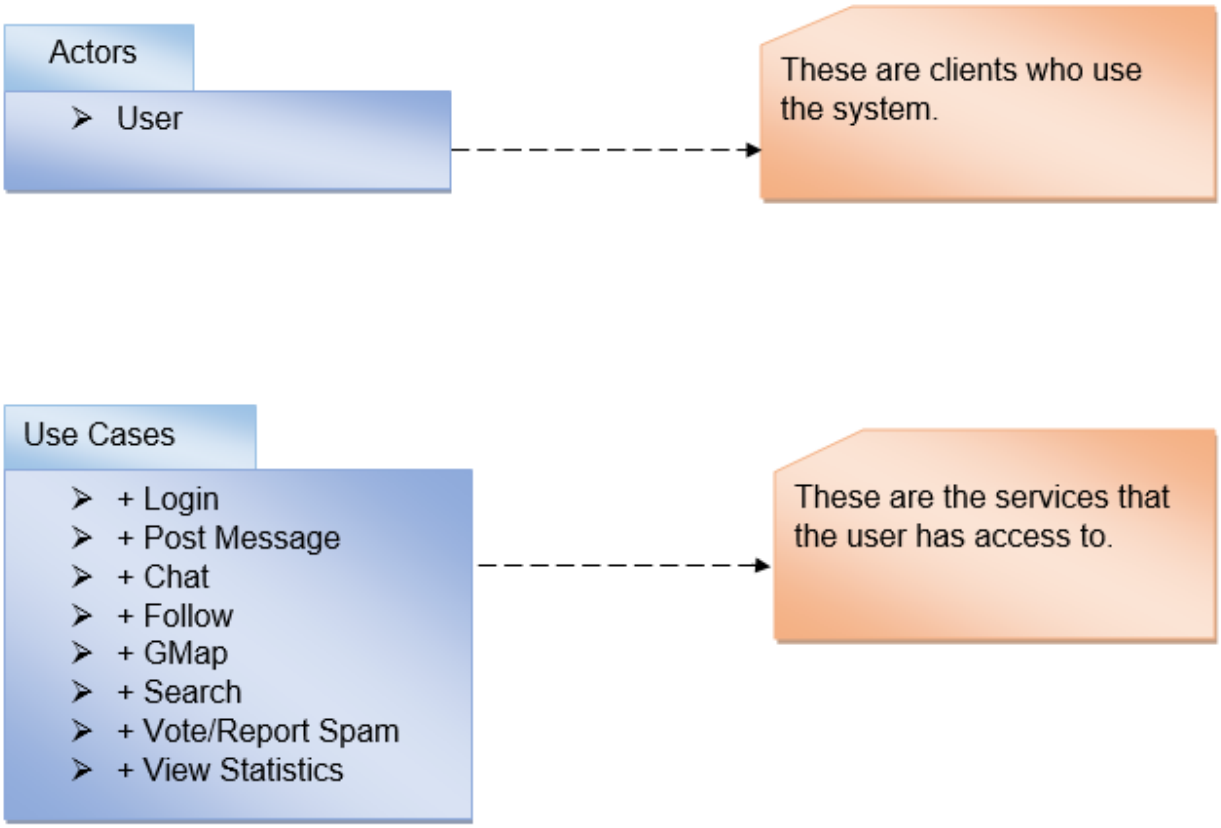


Figure 2 : General View

2. Use Case View

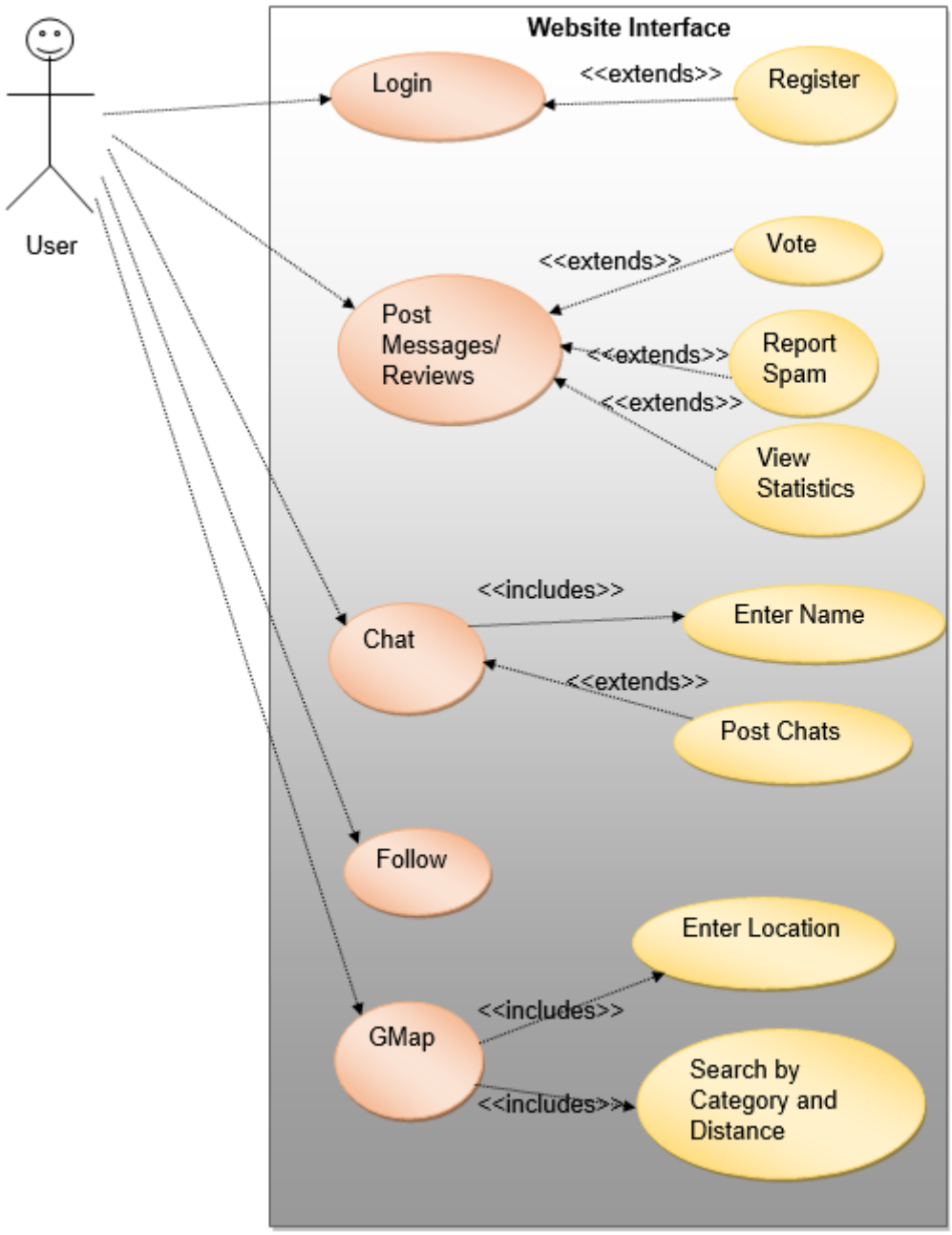


Figure 3 : Use Case View

4. Website Pages

The following pages are present in the website and are explained in order as below:

1. Login Page

This is the first page that the user encounters once accessing RePOST.

Technical Name: login.php

This page shows the username and password fields which the user has to enter correctly to get access into the website. After entering the correct credentials the user is taken to her/his home page if she/he clicks on log in button. If the user is a new user or a first time user, the user must click on the sign in button which will take him to the new registration page. The user must register on the website to use its services. She/he cannot view other members or access any of the pages contents from the outside.

2. Registration Page

This is the page that is displayed when the user clicks on the sign in button on the login page.

Technical Name: registration.php

In order to use the website's services the user must register with the website. For doing this, she/he needs to enter her/his username, email address, password for the website and a reconfirmation of the password. Proper validation checks are applied to ensure that the user enters all the necessary form data properly and does not fill in incomplete information. She/he has to click on signing up to finish the registration process and the user will be taken back to the login page where she/he will have to re-enter her/his credentials and gain access to the website's features. If the user clicks on the 'X' mark at the top of the sign in page widget then she/he is taken back to the login page.

Login and Registration Page Sequence

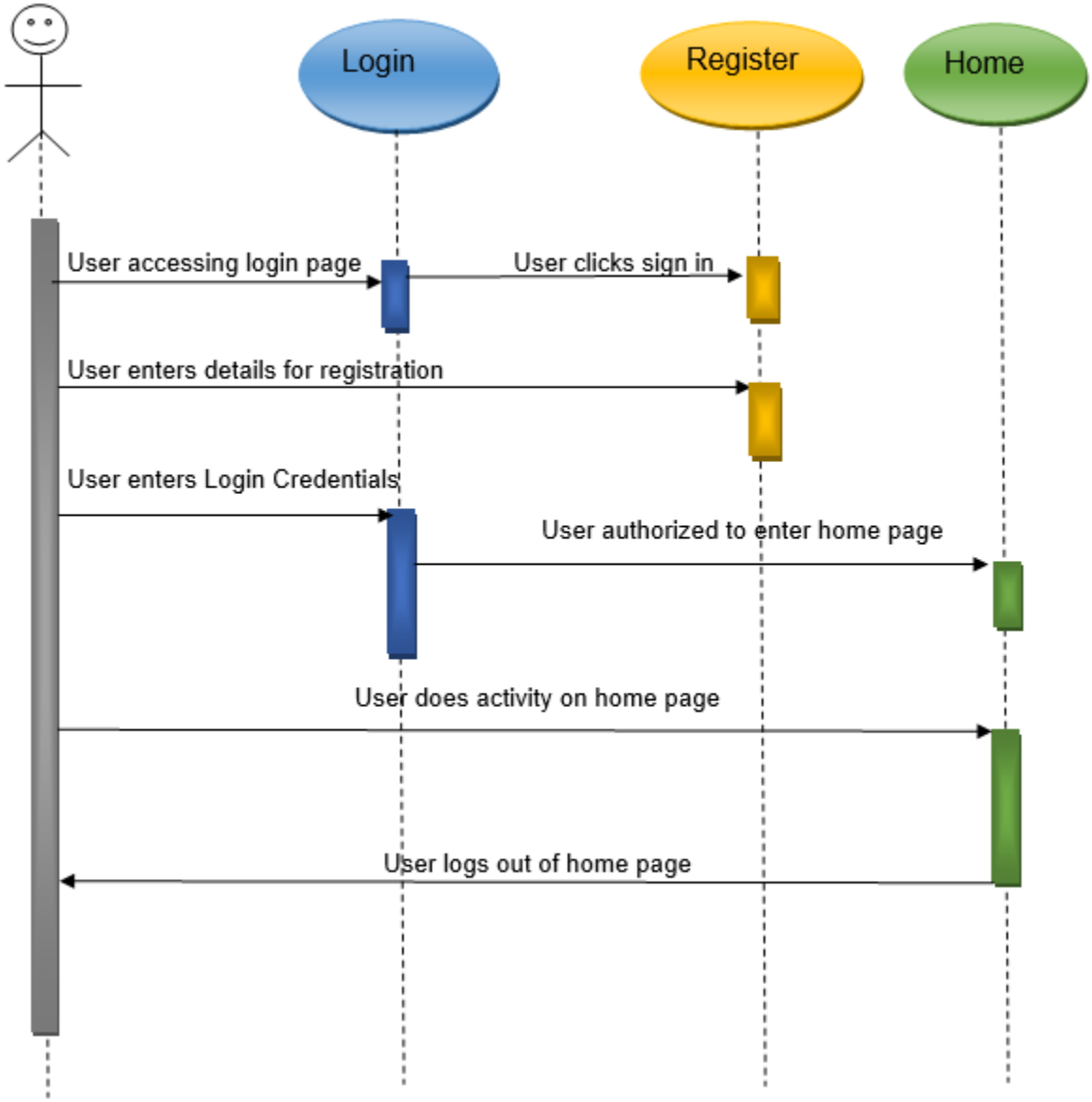


Figure 4 : Login & Registration Sequence Diagram

3. Home Page

This is the user's home page where the user is taken to after the login process.

Technical Name: index.php

Here, the user can view her/his posts, the posts of users which she/he follows and also a set of functionalities designed for the user. These functionalities are following other users, Google maps features and Chat feature. The user can post a message or a review by typing on her/his wall and clicking on post. This will post the message onto her/his wall and also publish her/his post onto the other users' walls who are following

that particular user. The user can also access the functionalities of Chat, Google Maps and Follow other users. There is a search bar in the top right corner of the user's home page. In this, the user can search for a particular post/message by Hashtags (#). E.g., there is a post with #SuperiorSandwiches in it, when the user inputs #SuperiorSandwiches in the search bar and clicks enter, all the posts and tweets which have #SuperiorSandwiches will be displayed.

Home Page Review Sequence

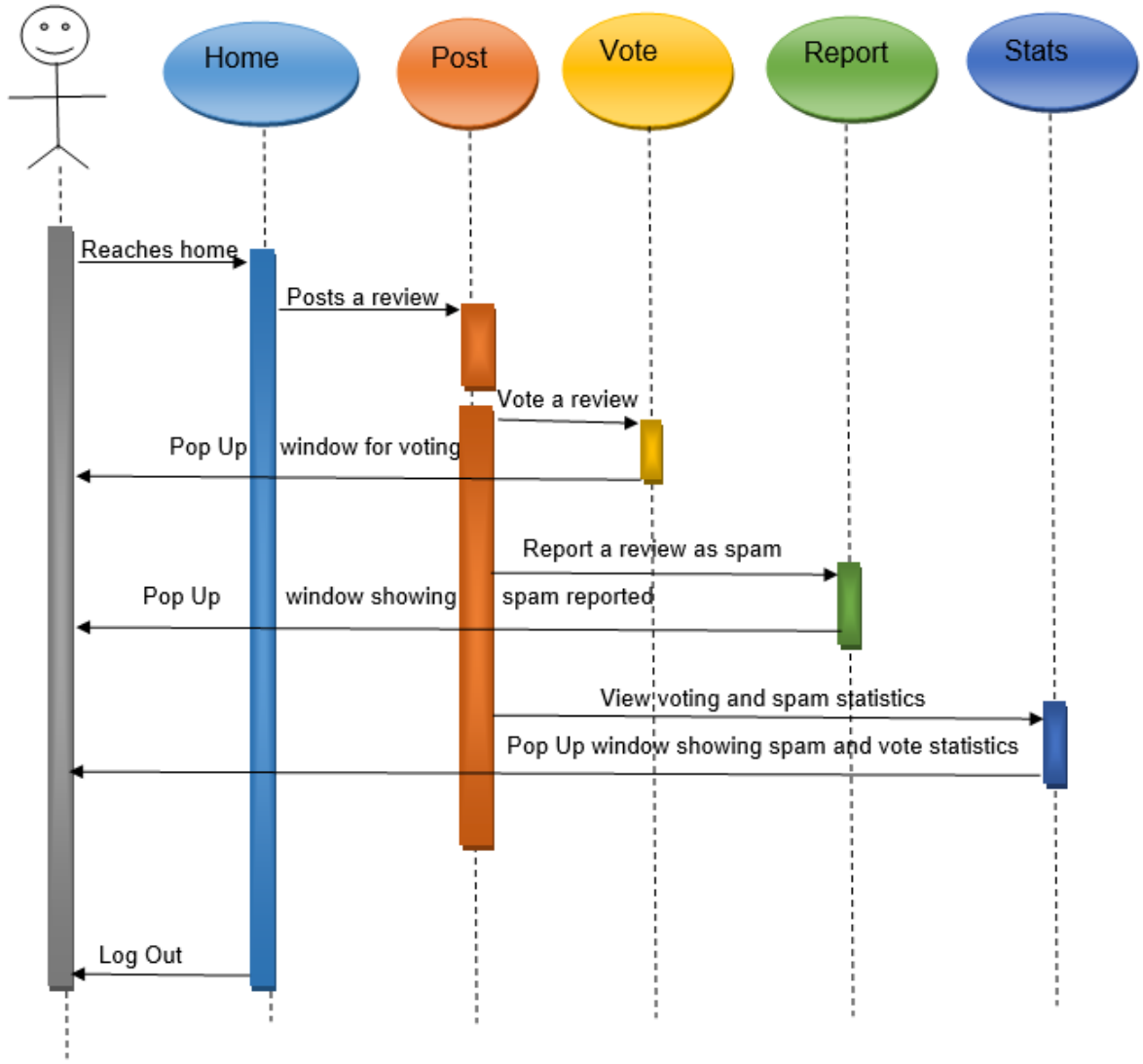


Figure 5 : Home Page Review Sequence Diagram

4. Follow Page

Here, the user can see and follow other users.

Technical Name: follow.php

The user on this page can see other users. Beside the other users, there is a follow button on clicking the user starts following them. The benefit of following is that the user can get the following user's posts on her/his wall. The user also has the option to unfollow a particular user.

Follow Sequence

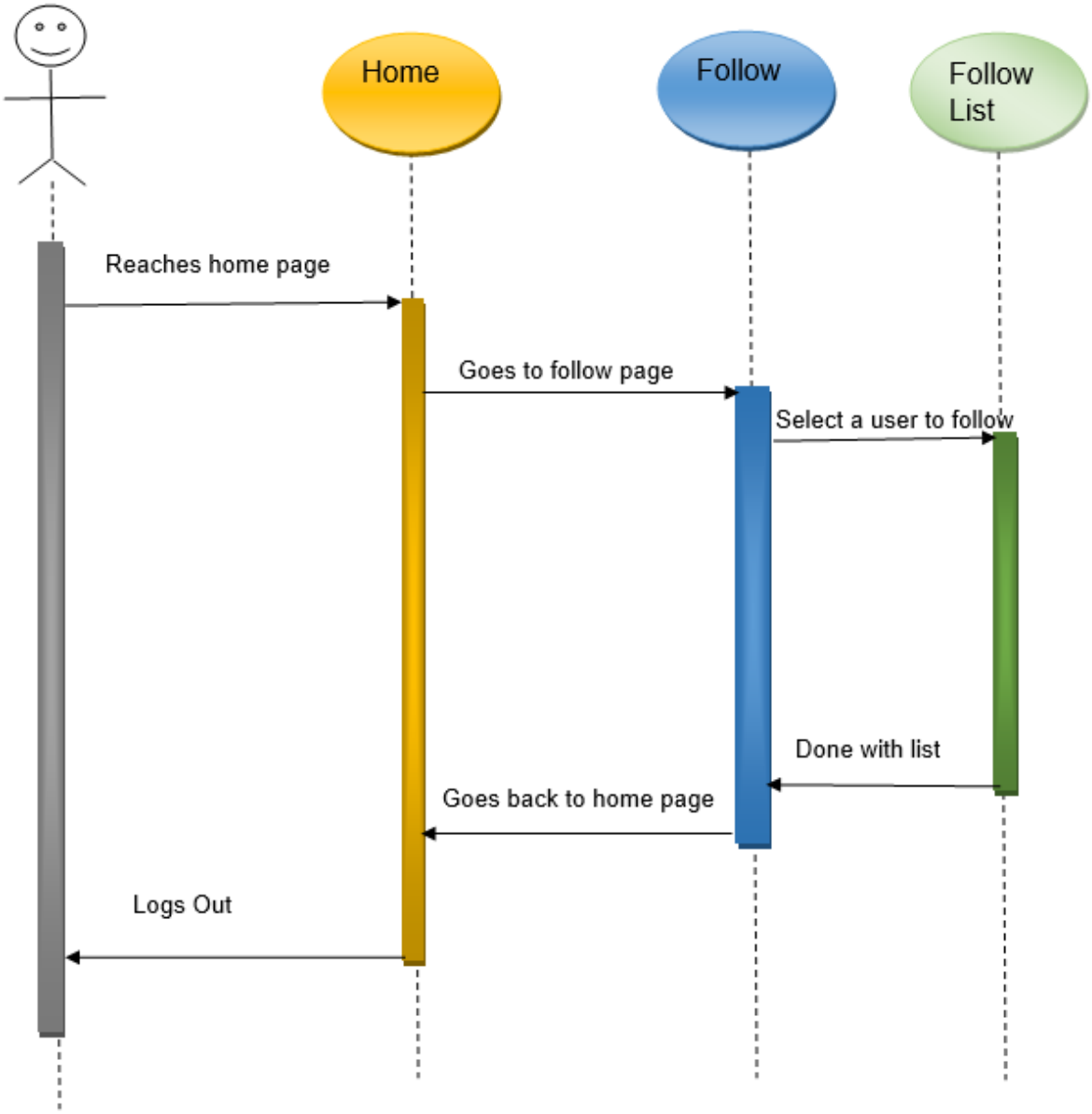


Figure 6 : Follow Sequence Diagram

5. Chat Page

Here, the user can chat with other users.

Technical Name: chat.php

The user clicks on the chat link on her/his home page and the user is prompted for a chat name. The user can enter a name of her/his choice and click ok to enter the chatroom. The name helps the user remain anonymous and she/he can chat with all other users who are present in the chat room at that time. The user enters the chat message and then presses enter on the keyboard to send the chat message. The chat message is displayed along with the username chosen is displayed. The chats also show the timestamp when they were posted for reference. The user can stop the chat at any time and then return to her/his home page. This chat is not persistent.

Chat Sequence

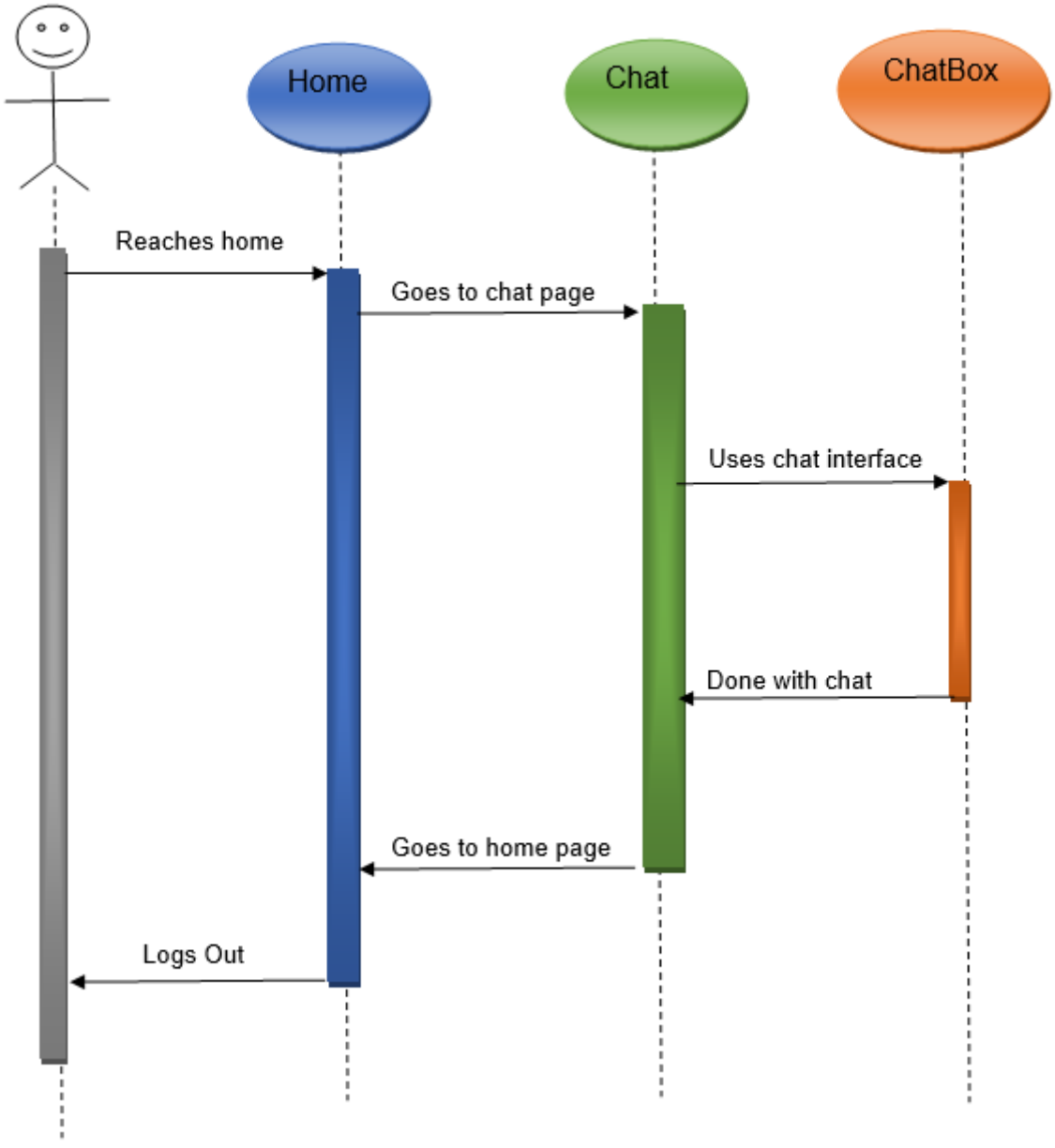


Figure 7 : Chat Sequence Diagram

6. GMap Page

Gmap allows users to search for places of their interest.

Technical Name: gmap.php

The user clicks on the gmap link on her/his home page and she/he is taken to the Google maps page on the website. This page contains a modified Google map which can be used by the user to search specific places of interest. These places are divided into the following categories: ATM, Bank, Food, Bar, Café, hospital, police and shopping. These places can be searched after entering a particular location where these places are to be found. There is also an option to search from what distance the places must be found from the location specified. The distances include 500, 1000, 1500 & 5000. This helps the user narrow down her/his search. E.g., the user enters Indianapolis as location the searches for it. A marker follows on Indianapolis center of the city and the user is taken there. After that she/he then selects Café and a distance of 1000 and enters as Starbucks. The gmap will show Starbucks in the area of 1000 from the marker placement. The user can thus search in a categorized manner and then return to her/his home page.

This works as follows:

1. When the user enters a location, selects a category, a fixed distance, and specifies the keyword [optional] these are passed as a function to the service function of Google Places API.
2. This API will then create the markers based on the request made by the user.

GMap Sequence

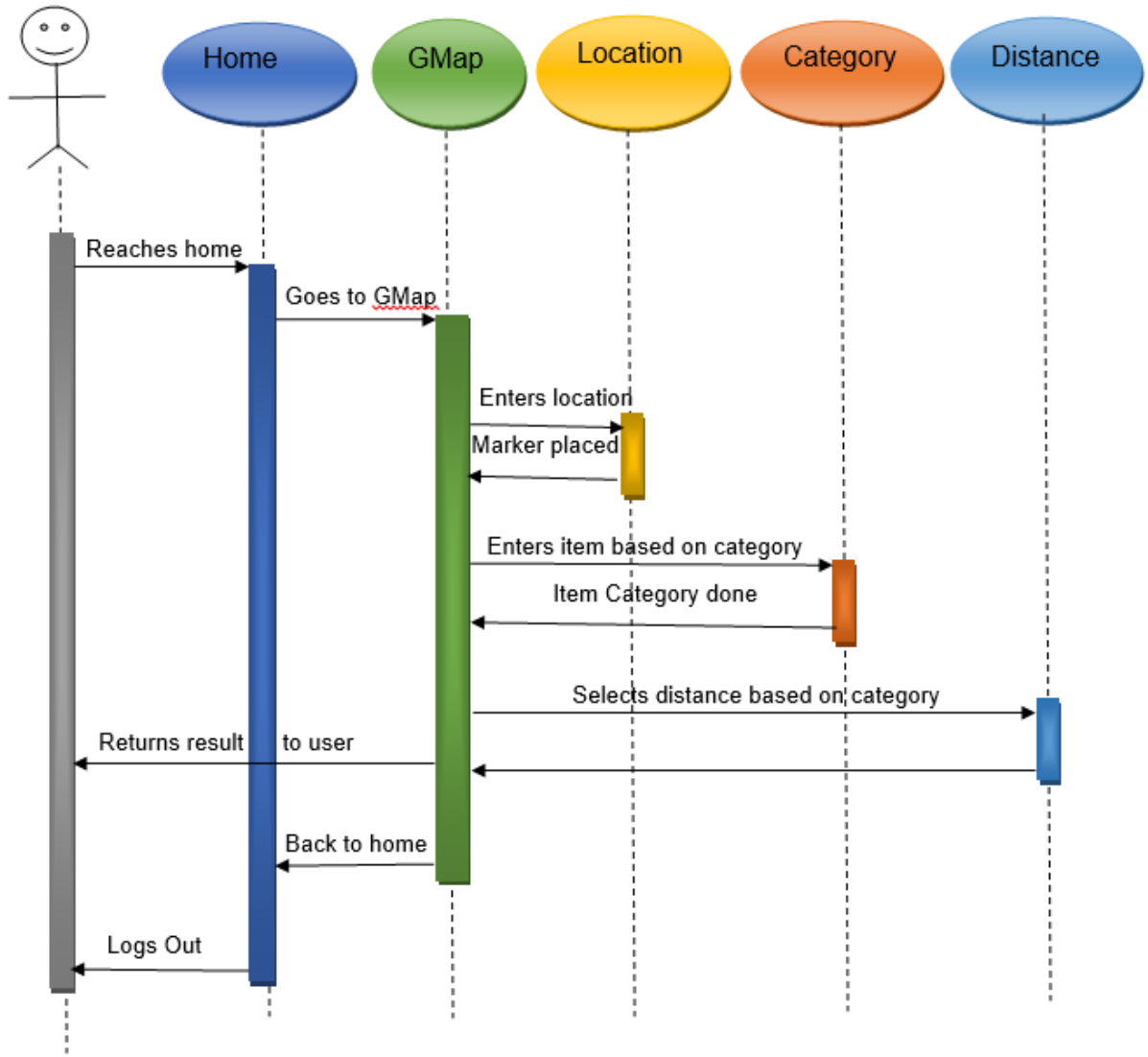


Figure 8 : GMap Sequence Diagram

5. Database Model of the system

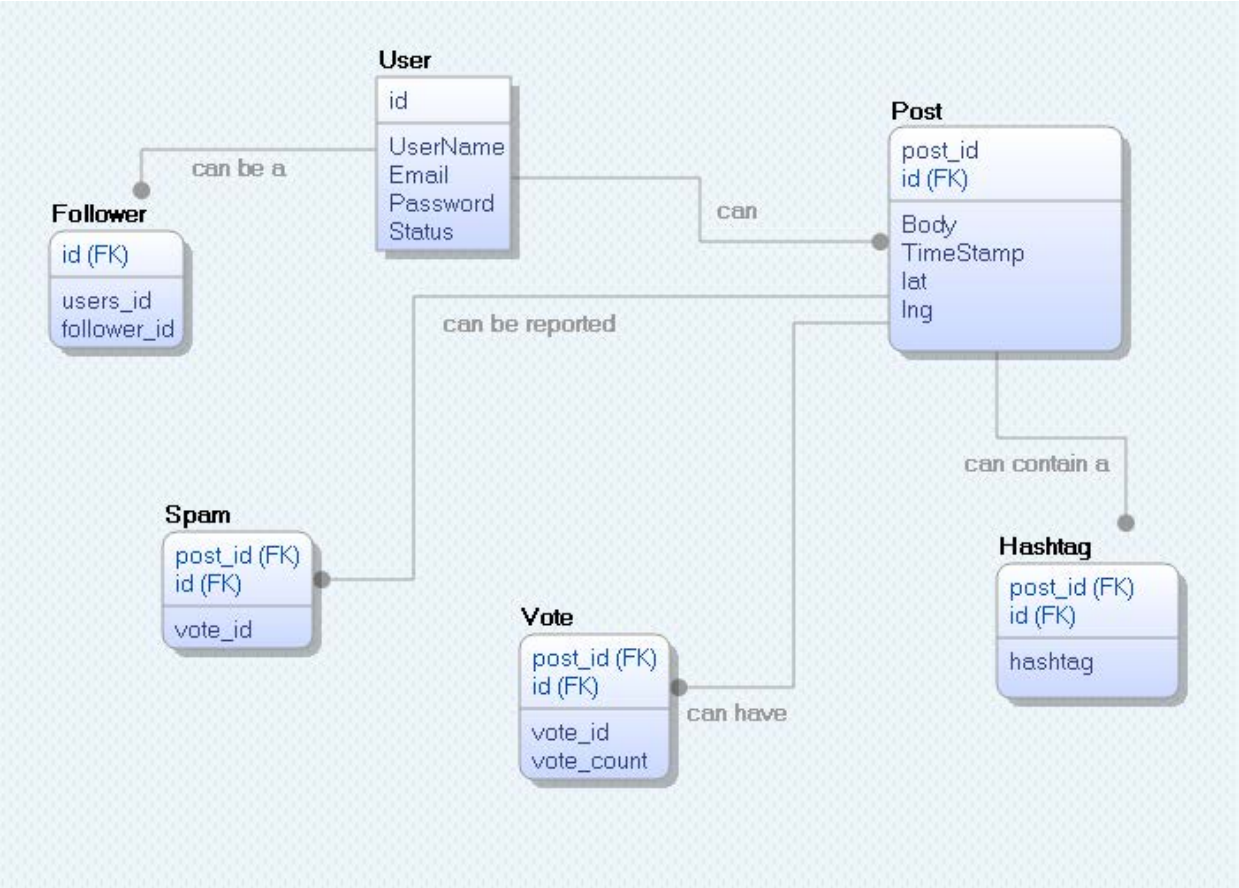


Figure 9 : Diagrammatic representation of the database using ERWin®

The main table in the database is the users table which stores all the information of the users. The passwords are encrypted and stored in the database for additional security. The status attribute is used to keep a check on if the user is active or not.

The post table stores all the data related to the various posts that the users post on their walls. It has a body field that is used to store the contents of the post. The timestamp is used to add a timestamp to the post. The latitude and longitude are used for location services.

The follower table keeps a track of which user is following whom.

The hashtag table is maintained to ensure that the search by hashtag feature works smoothly.

The vote table keeps a track of the ratings the users give on a particular post. One user can vote on a post only once. This has been done in order to reduce the number of false ratings on the posts.

A user can also report a post as spam. The spam table maintains a record of this. Again, a user may only report a post spam once. This has also been done in order to reduce the number of false spam reports on posts by bots or malicious users.

6. Additional Features of the Website

Posting a Review Features

The user can post a review/message on her/his wall in her/his home page. This will be displayed on others' walls who are following that user. Each of the other users have the following functionalities on the post. They can vote a post if they find that to be useful. They can also report the post as spam. This feature helps to remove fake reviews or messages. It also helps prevent negative or positive reviews on purpose. The users can view spam statistics for a particular post and also the vote statistics which shows the number of votes for that particular post. This will help the user judge whether a post is good or fake.

Vote Page

When the user clicks on vote for a particular review then a pop up window will open and the user can give a vote by selecting a rating from 1 to 5 stars. After selecting the desired rating the user clicks on vote which will register the vote. The system ensures that a person can vote only once. If a person tries to vote again then the message is displayed that she/he has already voted. This ensures that users can vote reviews only once and not abuse the system.

Vote Statistics Page

The users can view the vote statistics for a particular review by clicking the vote stats button. This will show the average rating based on the votes by the users. E.g., If 3 users vote a post with the rating of 5, 4, and 3 then the vote stats will show the rating as 4/5. This will thus show the popularity of the review and also indicate in a way the genuineness of the review. This will help other users know to judge based on the reviews.

Report Spam Page

The users if they feel that a post is not genuine they will vote as spam. This will ensure that purposely written good or bad reviews can be eliminated. This will help maintain the integrity of the website and the reviews posted by the people. When a user posts a review then if a particular user clicks on report spam button then a pop up window will open and it will show that spam has been reported.

Spam Statistics Page

The users can also view the spam statistics for a particular post by clicking on the spam stats button. This opens a new window and shows the users the number of times spam has been voted for that particular review. This will help the users to judge whether a particular review is good or just spam/not to be considered.

7. Deployment on the cloud

We deployed our project on the cloud so that other users can access the website easily and efficiently. We chose to work with Amazon EC2 cloud to deploy our project due to its flexible nature and ease of usage.

Part 1: Creating an instance/virtual machine on the cloud

1. The first requirement is to create an instance on EC2. The instance we created and used is based on Ubuntu AMI [Amazon Machine Image].
2. Next we need to select an instance type. The instance selected.
3. The third step involved is adding 'Configuration Instance Details'. The default options were selected on this page.
4. Step 4 involved creating and adding a storage associated with the instance. The standard storage EC2 provides is 8 GB.
5. In the next step we were asked to tag our instance. We left this screen blank and took the default options.
6. For security groups we used the following options:
   - ✓ SSH with TCP protocol and port range 22
   - ✓ HTTP with TCP protocol and port range 80
   - ✓ HTTPS with TCP protocol and port range 443
   - ✓ Custom TCP rule with port Range 8080

   These particular rules are selected in order to gain access to the instance from our desktop, transfer our files and then deploy our project on the cloud.

Part 2: Installing the required software on the virtual instance

In part 1 we created our virtual instance but this is 8 GB of space with no software installed on it. We now need to install the necessary software on this instance to deploy our project. The software we installed are PHP, MySQL and MySQL server on the virtual machine. The process of the same is described below.

1. Just before creating the instance EC2 asks us which key pair we intend to use with the virtual machine or we want to create a new key pair. One instance can generate a key pair only once and hence this should be stored with care. We then proceeded to create a new key pair as a key pair had not been generated for this instance before.
2. After generating this key pair we downloaded Cygwin for windows. Cygwin gives windows a UNIX like shell to work in and this is needed to SSH into our instance which is running Ubuntu.

3. Cygwin was linked to work with CMD by changing the environment variables and adding C:\cygwin64\bin at the end of the currently set environment variables.
4. After this we need to connect to our virtual machine in the cloud via our desktop. We used the following command to connect:
   ssh –i Filezilla.pem ubuntu@ (*insert current IP address on which the instance is running)*
   Filezilla.pem is the name of the key pair we created.
5. This command enables us to create a connection between our machine and the virtual machine.
6. Next we need to install the various software on the machine that we need to deploy our project. The software we installed were PHP, MySQL server.

Step 3: Transferring files to the cloud

The penultimate step was to transfer our files to the cloud. We used Filezilla to achieve this. Filezilla was chosen as it provides an easy to use GUI.

1. In Filezilla, the first setting that was changed was to add the key that we generated from our instance.
2. Once this was done, we could simply connect to the cloud and drag and drop our files.
3. The folder where we added our files is /var/www.

Step 4: Database deployment on the cloud

The last step was creating a copy of the database on the cloud as we had on our local machine. We installed MySQL on our virtual machine to allow us to achieve this. We created our database again by entering the appropriate commands for creating the various databases/tables and with this our project was completely deployed on the cloud.

8. Benefits of Deployment on Cloud

There are various benefits for deployment of our project on cloud. They are listed below:

1. Minimal Cost

The cloud deployment costs lesser as we do not need to worry about buying servers and domain names and the server installations etc. We just need to buy the credits and use them and will be charged on the website's usage. So, initial installation cost is eliminated.

2. Focus on Website

We can just focus on the actual code and structure (software) of the website leaving aside the housekeeping and additional maintenance (hardware) of the website. The hardware and maintenance of servers work will be done by amazon web services and we can focus on building our website.

3. Speed of Deployment

As we experienced to go live the project could be deployed very easily and fast on cloud. This is much faster than actual server installation.

4. Ease of Use and Management

The cloud deployment can be easily used and managed. We just need to spend money as per our usage and only take care of the software aspect of the website i.e., the website's code and not worry about the server's functionality.

5. High Scalability

Cloud deployment can be scaled easily and that is taken care of by the Amazon Web Services and they will allocate proper resources and bandwidth as per our usage. This would have to be done manually if we had not deployed on cloud. We just need to pay as per the usage of resources and nothing else.

6. Better Business Model

Normally, we would have to ourselves spend money and resources to initiate the process of installation and maintenance but here we just need to pay as we use. So, the initial cost is eliminated and if our site grows we need not worry about details like changing our servers and increasing the bandwidth as that is taken care of by AWS.

7. Redundancy and Resilience

The website will work always and have good uptime as AWS will take care of the same. Otherwise we would have to worry about all these details.

8. Legal Contract

As we have lawfully paid AWS and have a good contract, AWS will ensure facilities like 100% uptime, proper bandwidth, scalability, accessibility etc. for our website. Hence we chose AWS.

9. Technologies Applied

This section describes the technologies/tools used during the development of RePOST.

1. Technologies

Front End: HTML. CSS, Java-script
Back End: PHP, MySQL

2. Development Environment

Eclipse for designing the website
Erwin for database design
XAMPP server for local testing
Amazon EC2 Cloud – For cloud deployment

10. Screen Shots

To illustrate the use of our website on multiple devices we have two users Arvind and Pulkit. Arvind is using the website on his Windows laptop and Pulkit is using his Android cell phone. Since there are no users in the database at this point of time, we have started the process with registration.

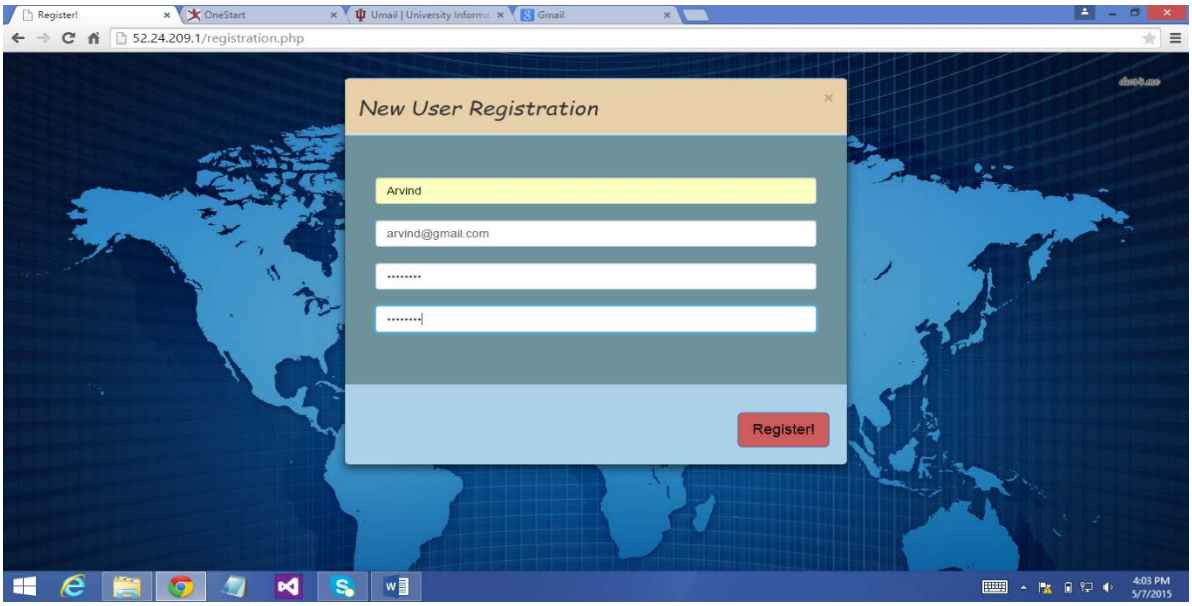
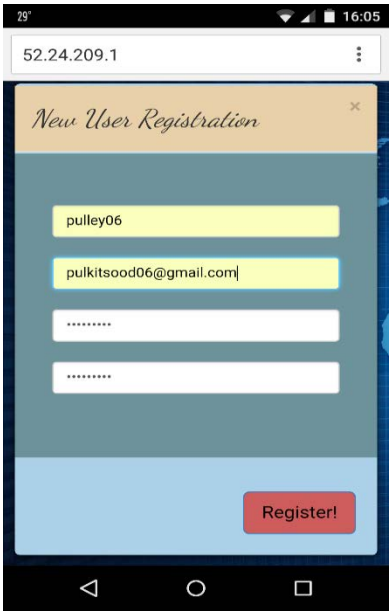


Figure 10 : Registration Page for PC



The image above is taken on the Desktop and the image on the left is the screen shot taken from the android device.

The user needs to enter all the fields else an error message is displayed. The password needs to be a minimum of 8 characters and is encrypted with MD5 hashing to provide security and to ensure that they cannot be accessed illegally.

Figure 11: Registration page on Android

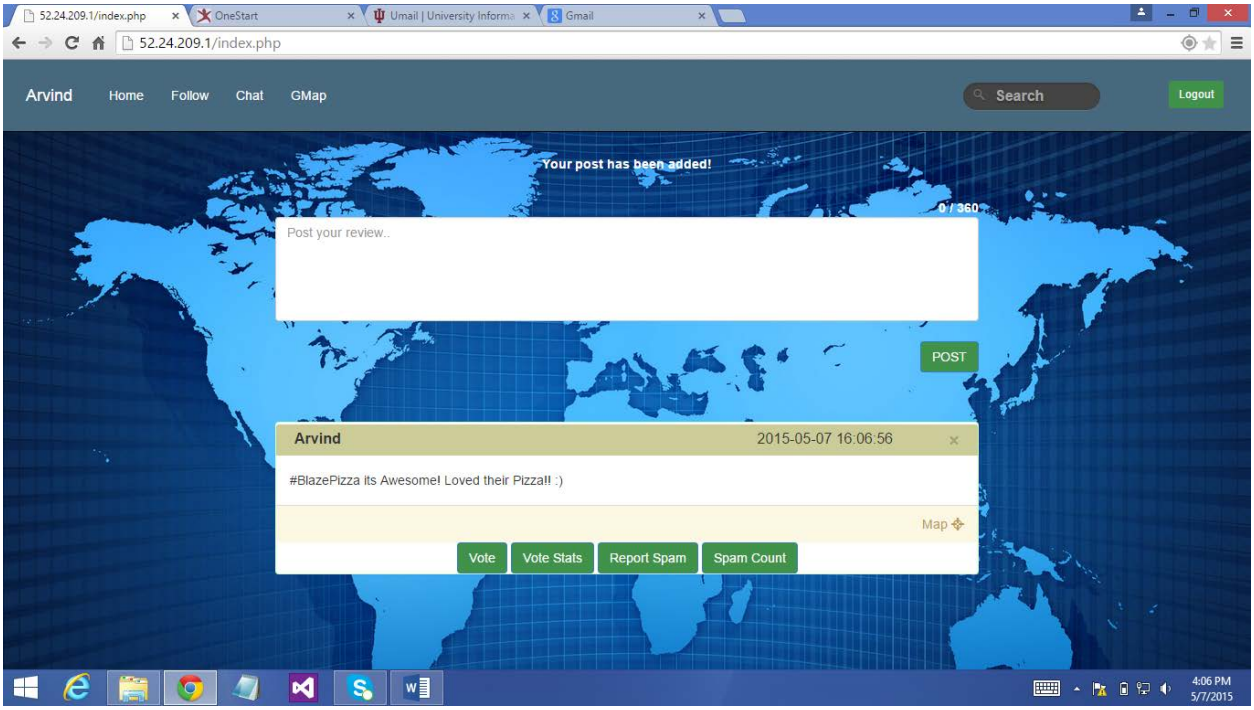
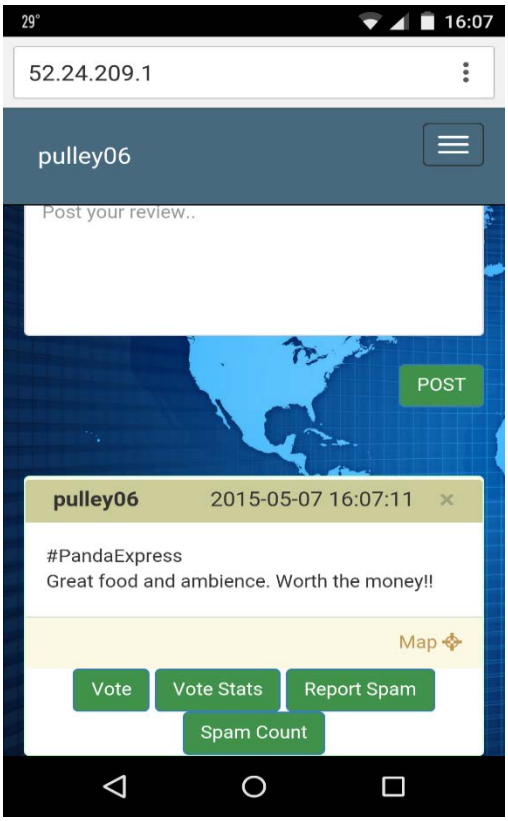


Figure 12: First post for PC User



In the screen shot above the user makes his first post on the PC. In the image on the left the user makes his first post on the Android device.

Figure 13: First post for Android user

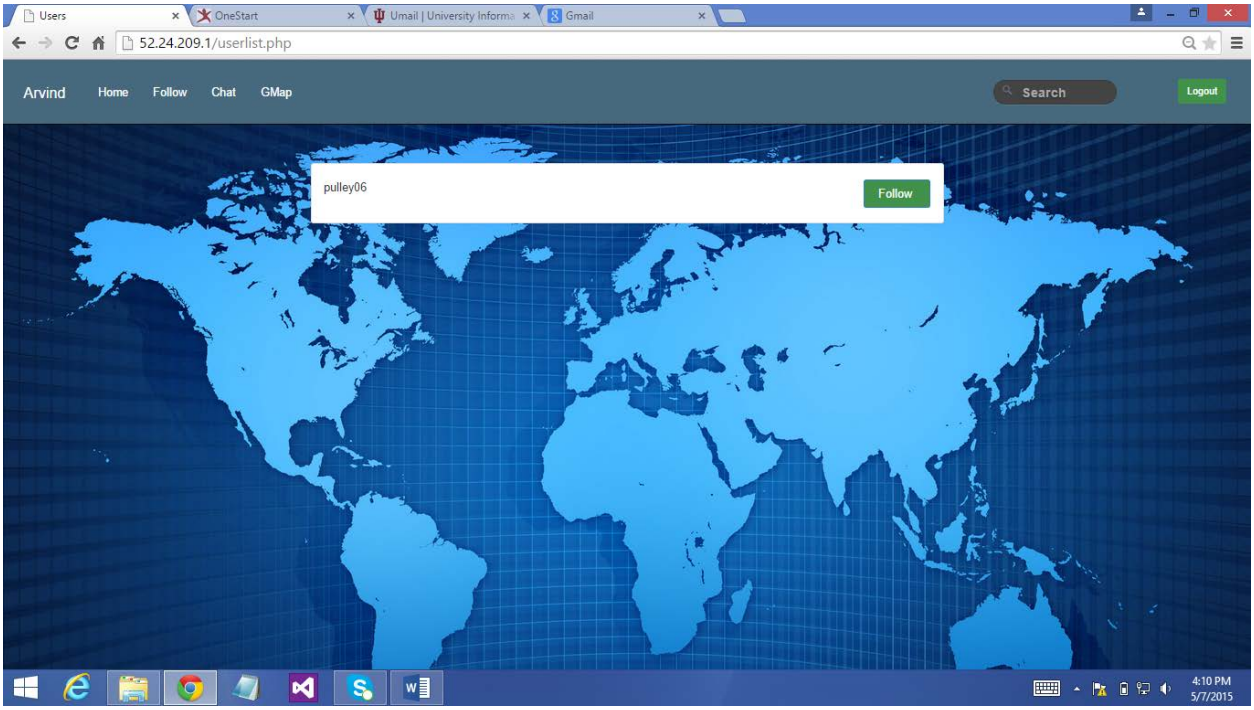
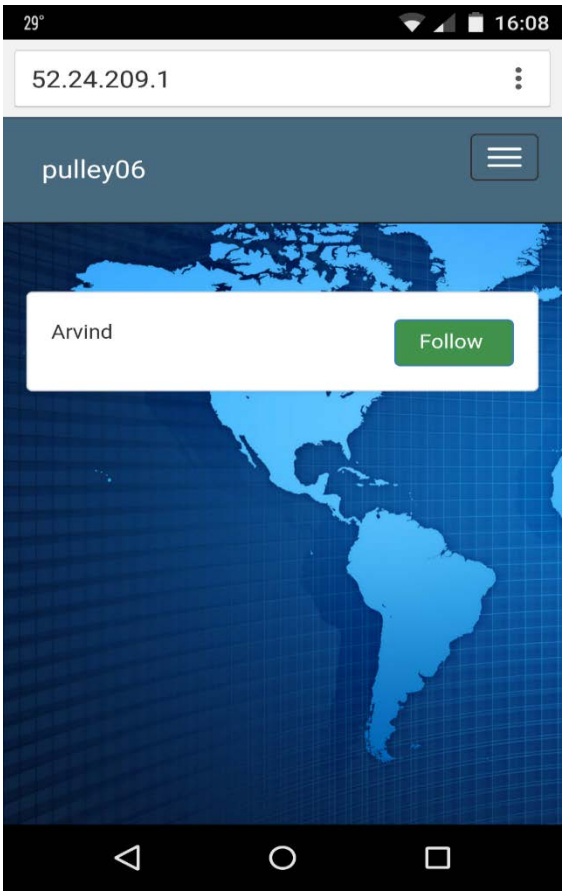


Figure 14: Follow page on PC



In the screen shot above the user wished to follow another user and clicks on follow on the PC. In the image on the left the same is done on an Android device.

Figure 15: Follow page for Android

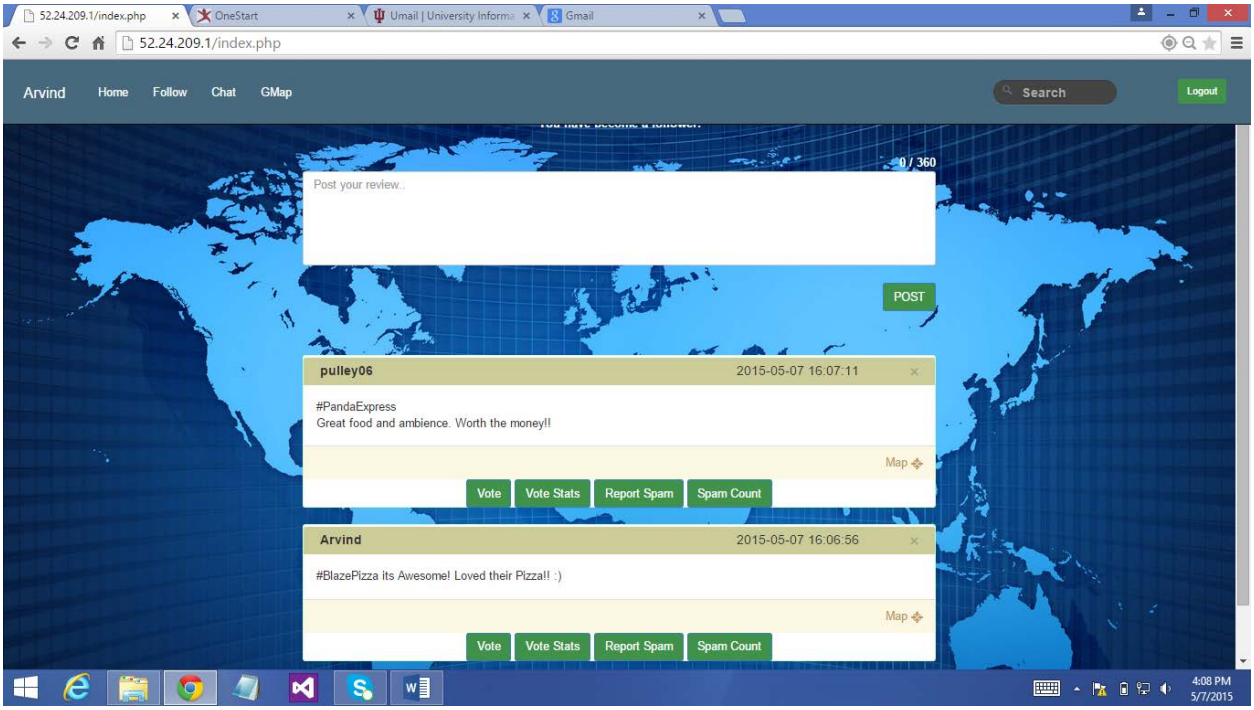


Figure 16: Wall after follow on PC

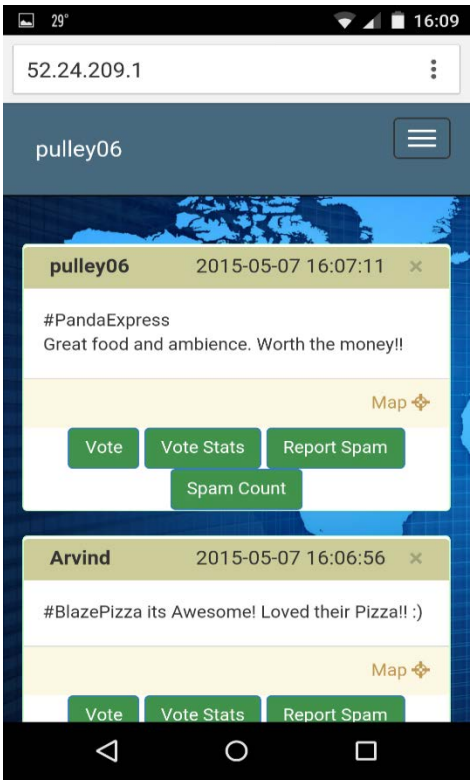


Figure 17 Wall after follow on Android

In the screen shot above the user's wall is displayed after following on the PC. In the image on the left the same is shown on an Android device.

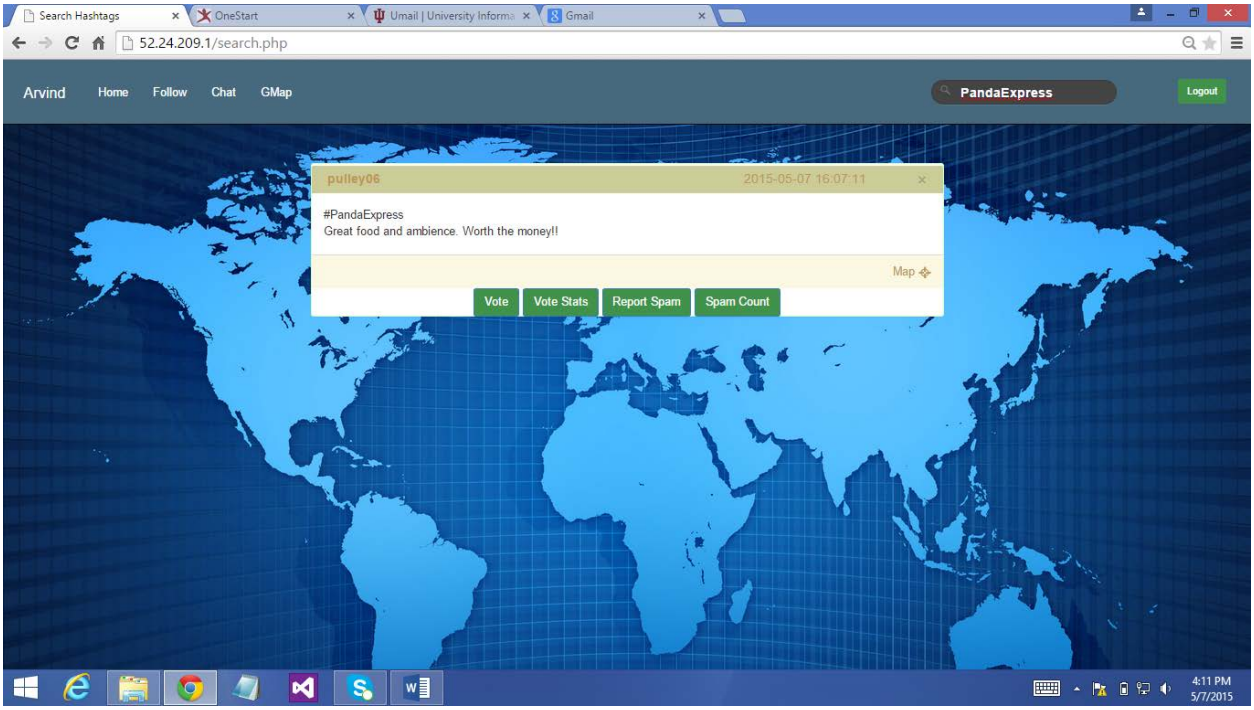


Figure 18: Search hashtag on PC

In the figure above, the user searches for a hashtag on PC & below on Android.

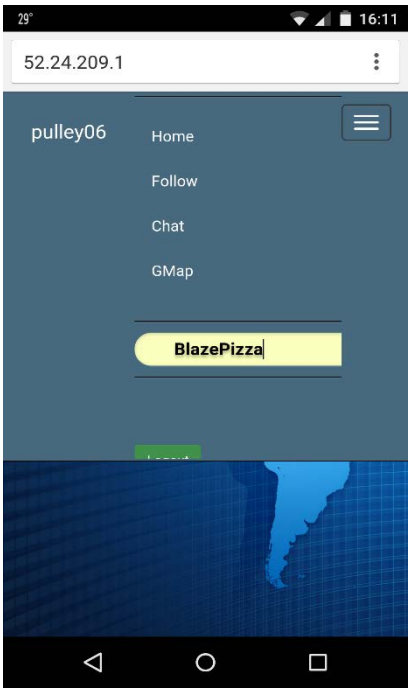


Figure 19: Hashtag Search on Android

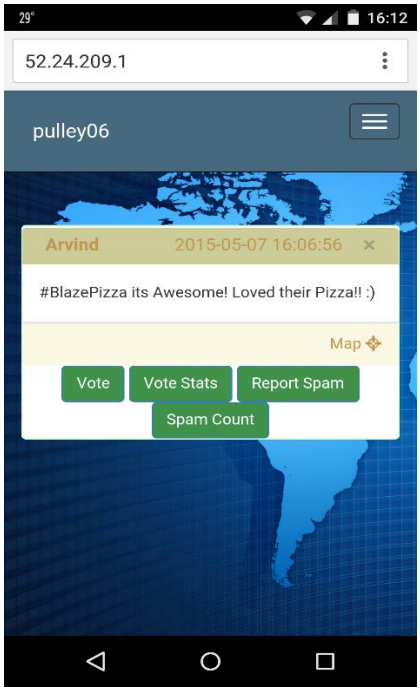


Figure 20: Hashtag Search on Android

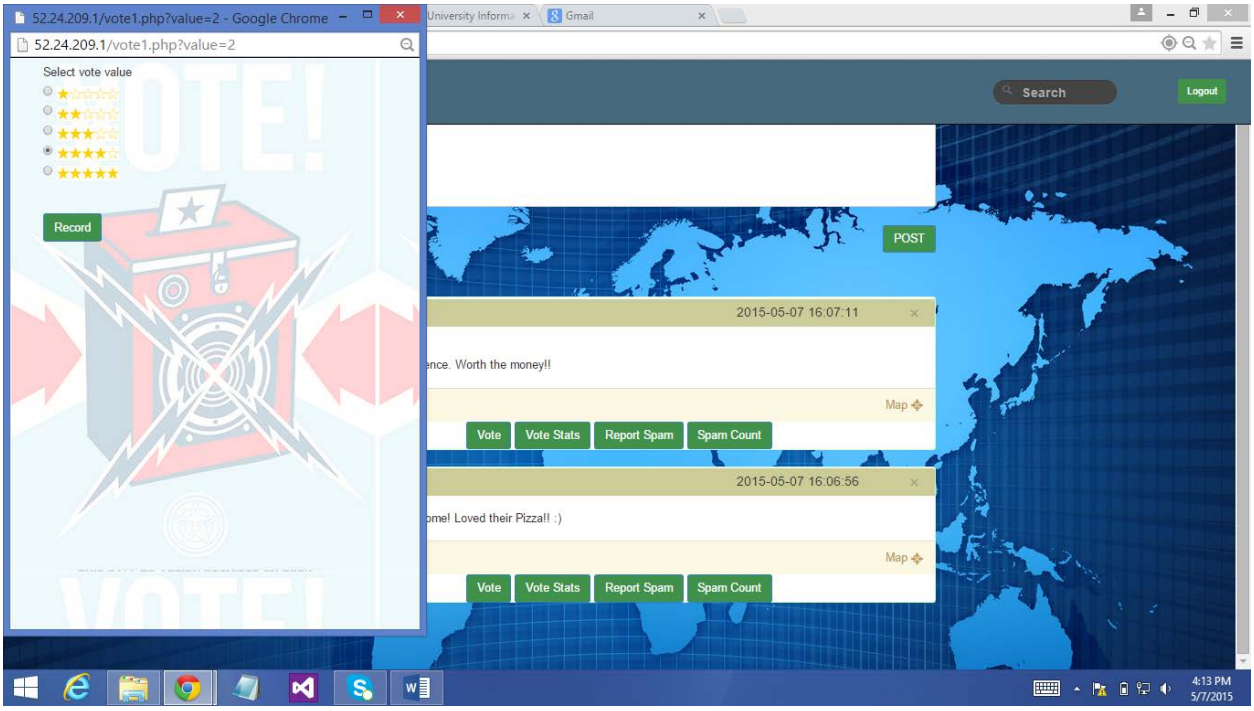
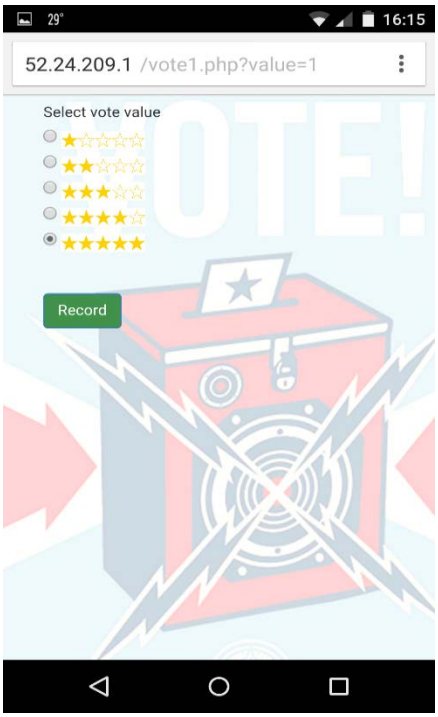


Figure 21: Voting on a review on PC



In the screen shot above the user can choose a rating from 1 star to 5 stars for a review on the PC. In the image on the left the same is shown on an Android device.

Figure 22: Voting on a review on Android

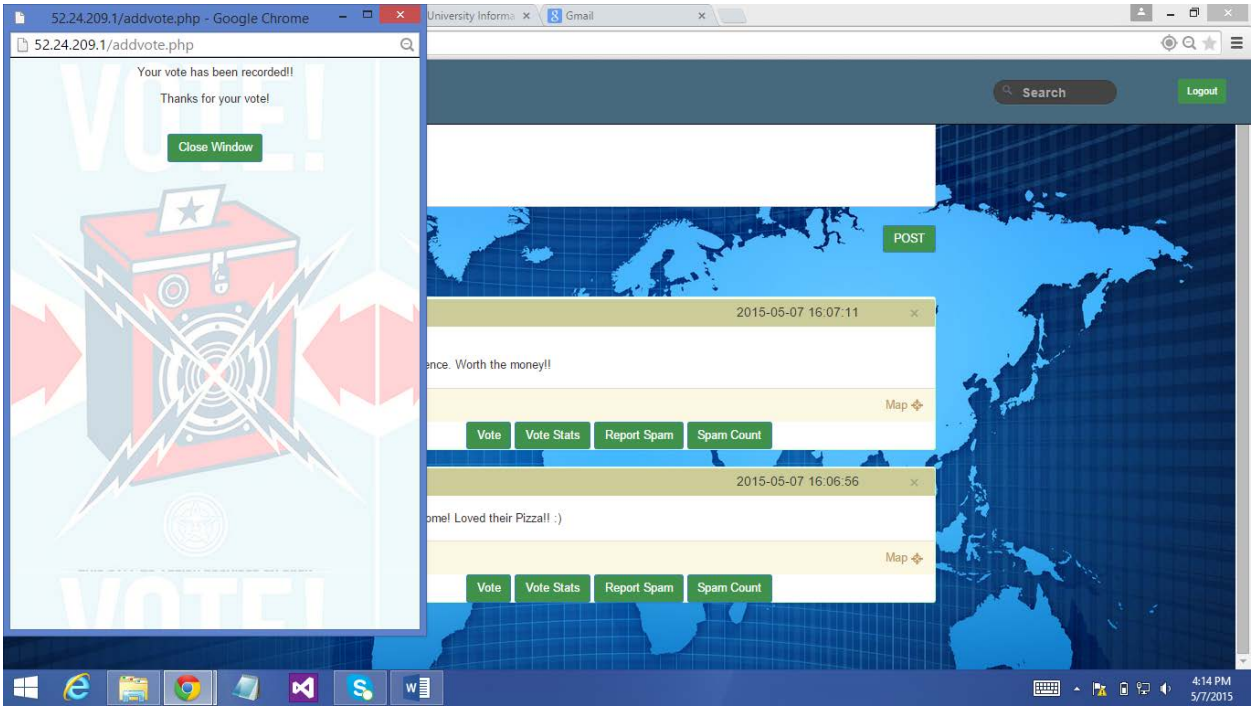
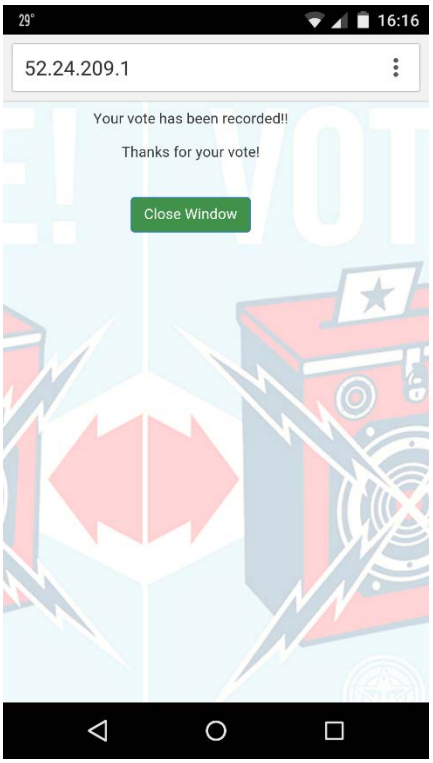


Figure 23: Confirmation message after vote on PC



In the screen shot above the confirmation message after voting on a review is displayed on the PC. In the image on the left the same is shown on an Android device.

Figure 24: Confirmation message after vote on Android

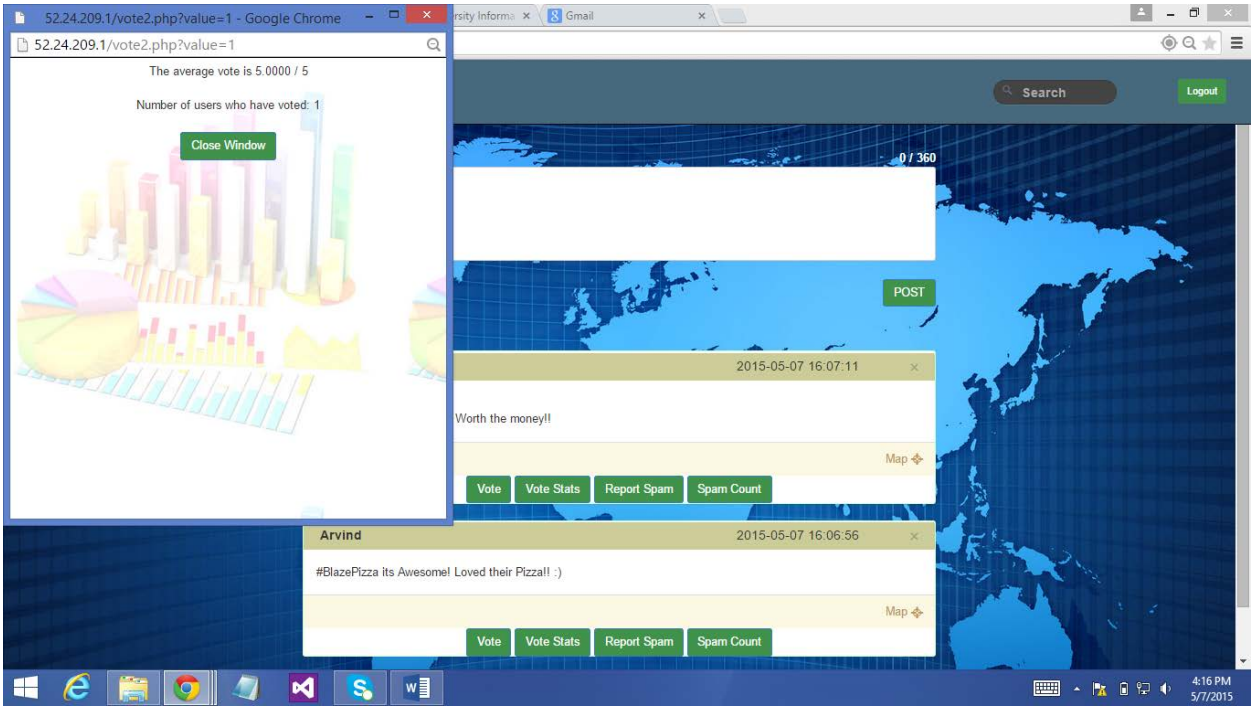
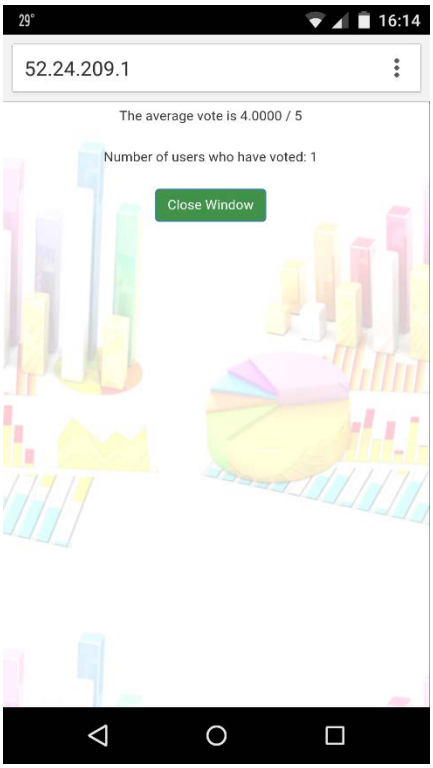


Figure 25: Vote stats for a review on PC



In the screen shot above the vote stats are displayed on the PC for a review. In the image on the left the same is shown on an Android device.

Figure 26: Vote stats for a review on Android

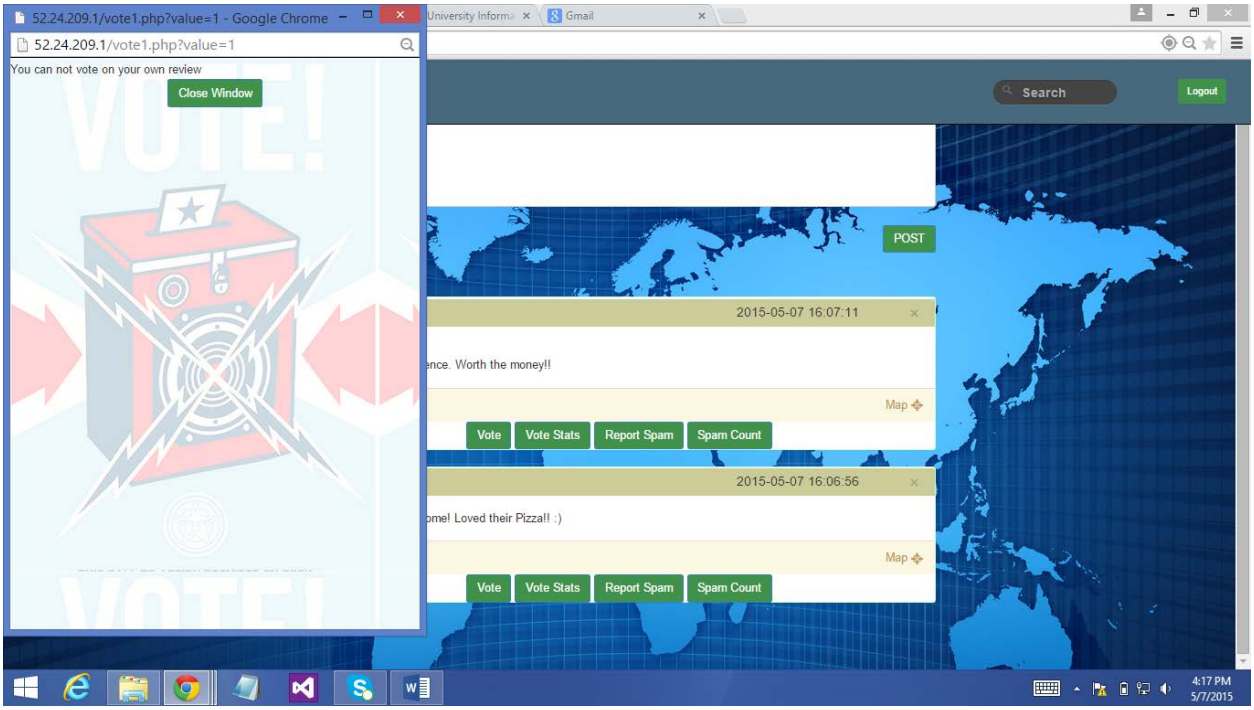
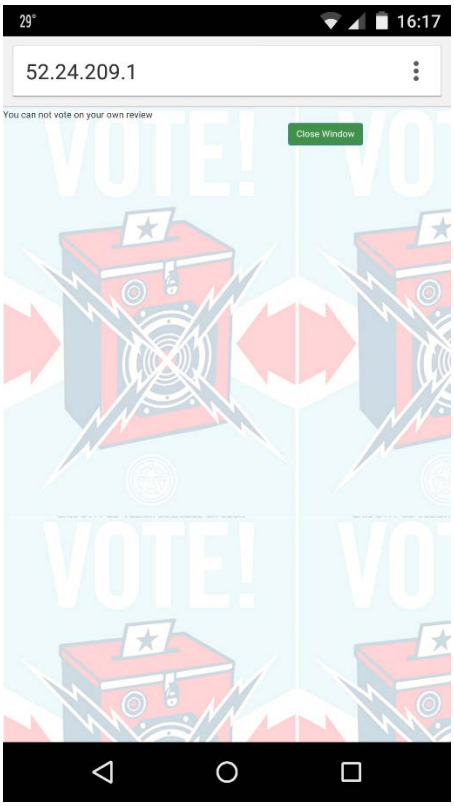


Figure 27: Error message for self-vote on PC



In the screen shot above an error message is shown on the PC when a user tries to vote on his/her own review. In the image on the left the same is shown on an Android device

Figure 28: Error message for self-vote on Android

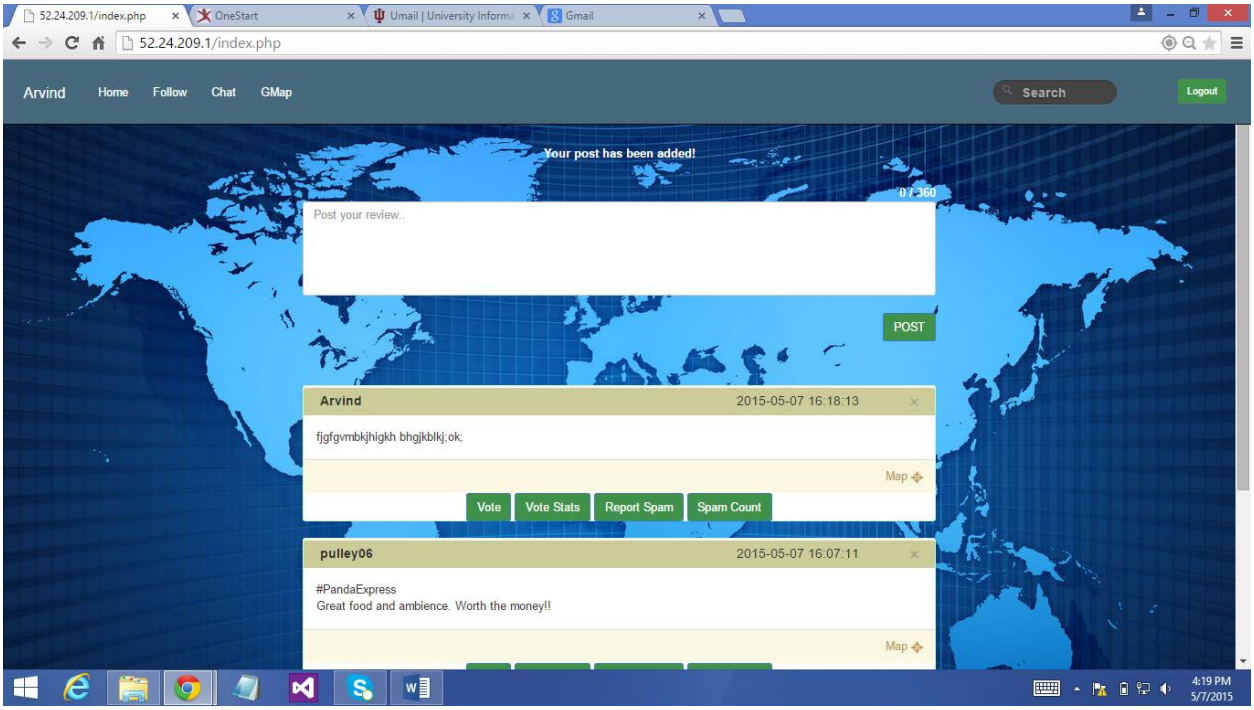
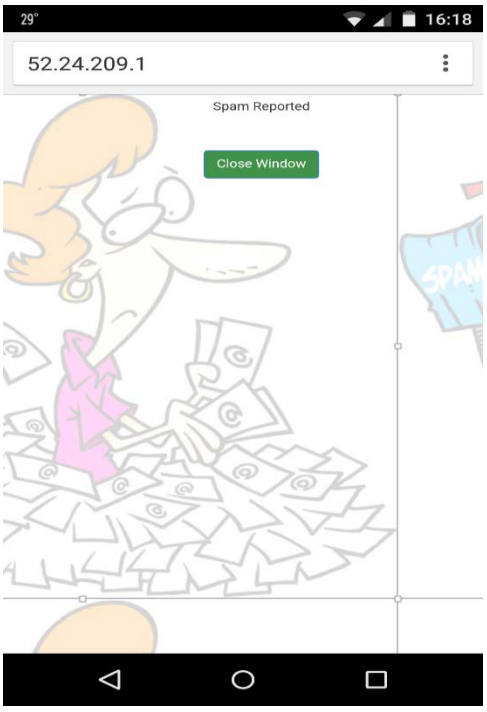
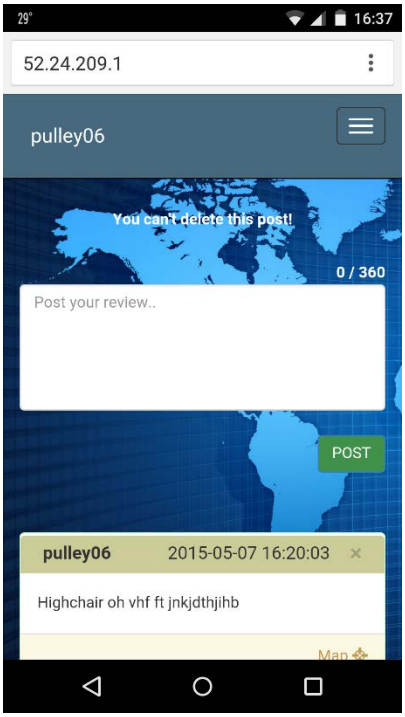


Figure 29: User Arvind has posted a spam message



In the screen shot above the user Arvind posts a spam message. In the image on the left user pulley06 reports the same as spam.

Figure 30: User pulley06 has reported this as spam



In the image on the left user pulley06 posts a spam message on Android. In the image below Arvind reports it as spam on the PC.

Figure 31: User pulley06 posts a spam review

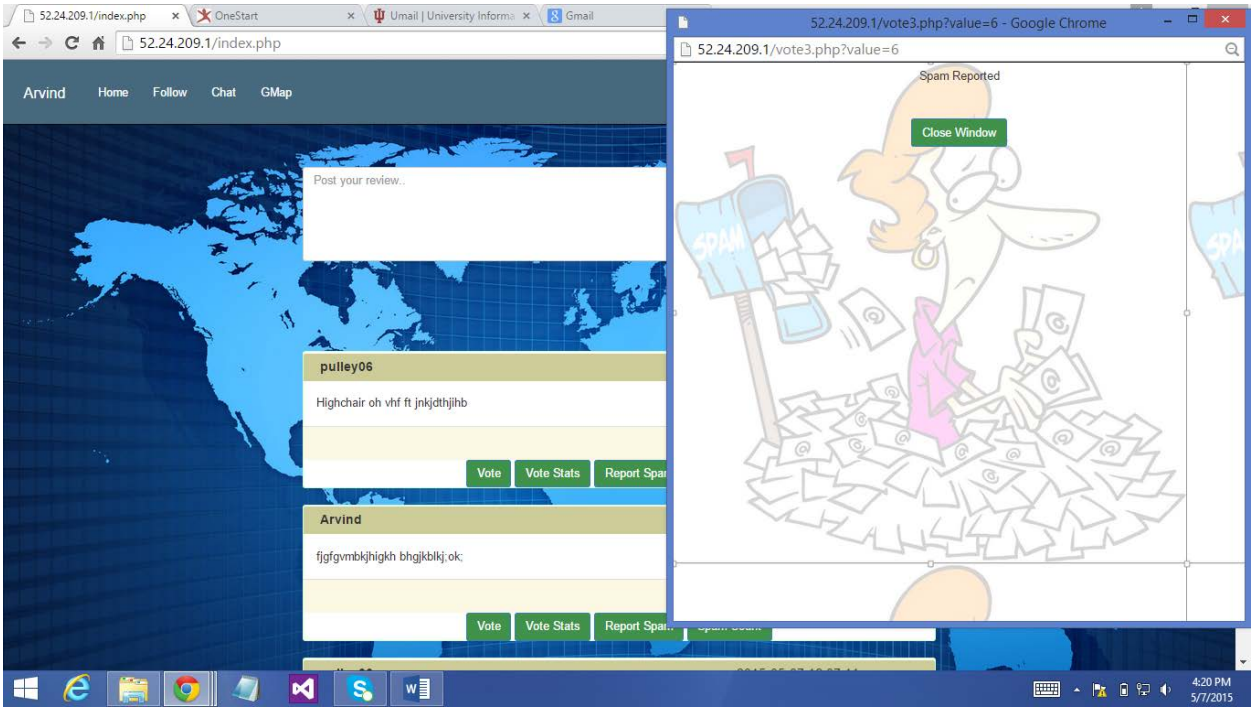


Figure 32: User Arvind has reported this post as spam

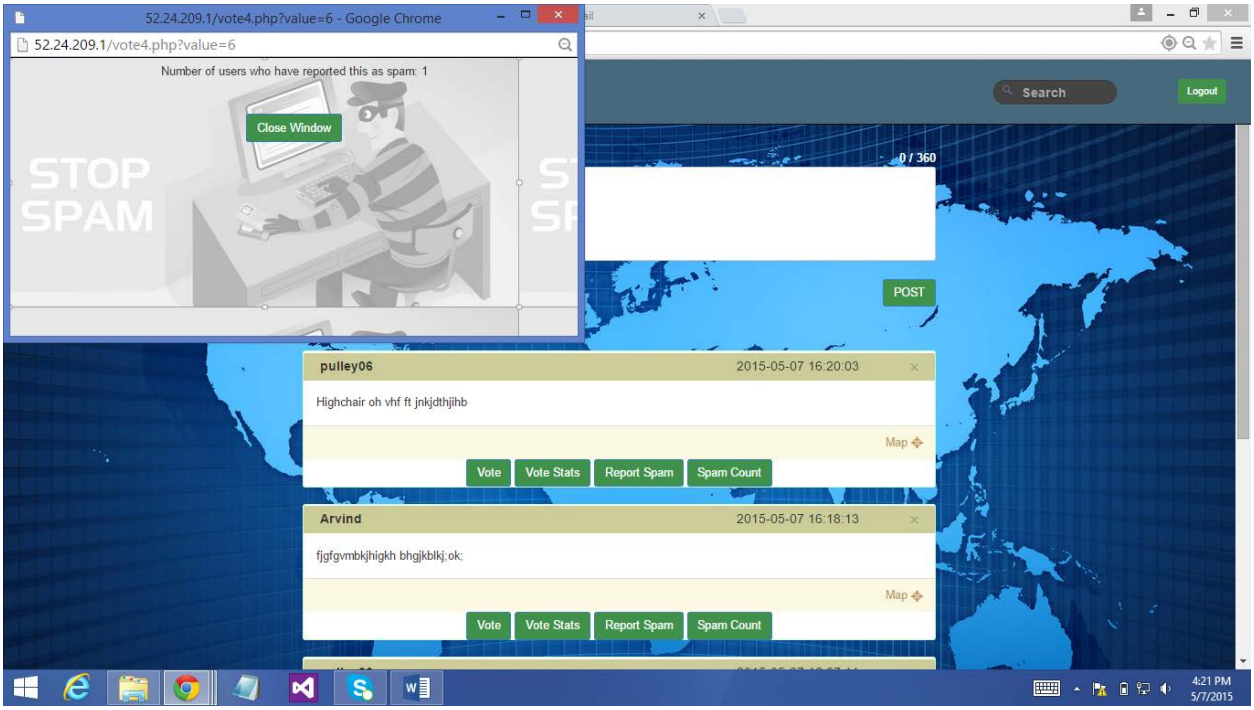


Figure 33: User Arvind views spam count on PC

In the screen shot above the user Arvind views spam count for pulley06's post on the PC. In the image on the left the vice-versa is shown on an Android device

Figure 34: User pulley06 views spam count on Android

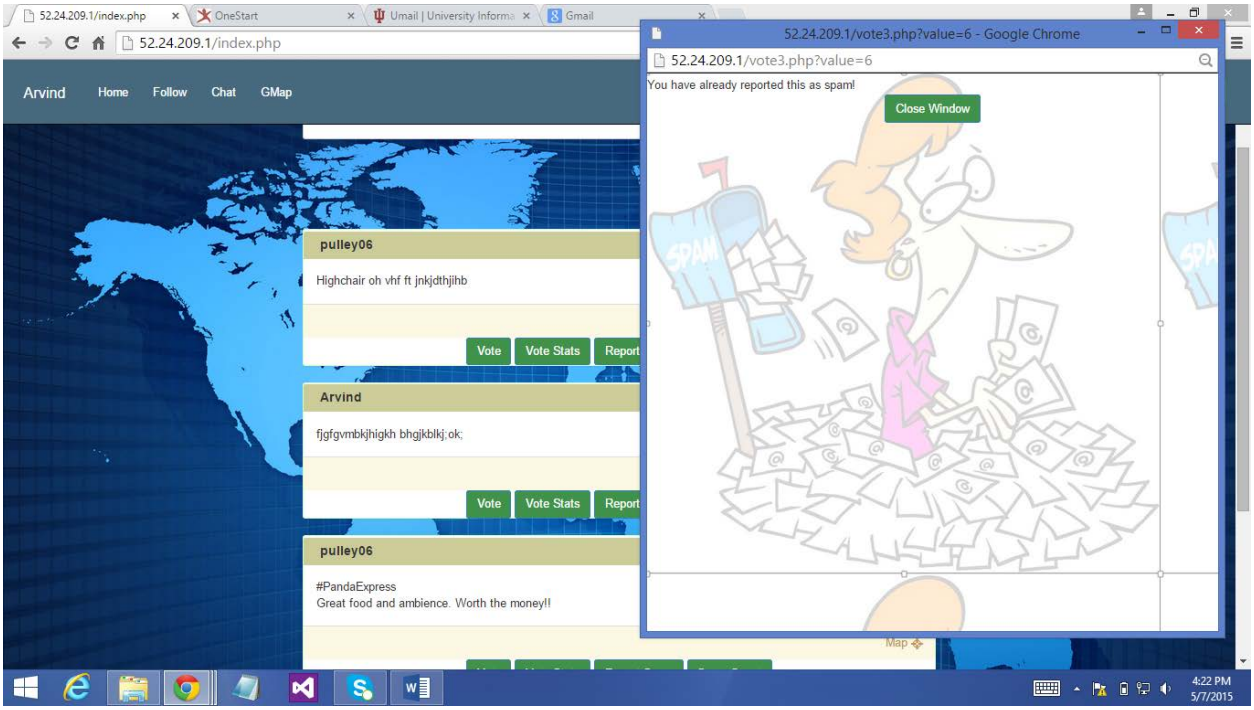


Figure 35: User Arvind tries to report review as spam again

In the screen shot above the user tries to report a review as spam after already reporting it once. Hence an error message is displayed. In the image on the left the same is shown on an Android device

Figure 36: User tries to re-report spam

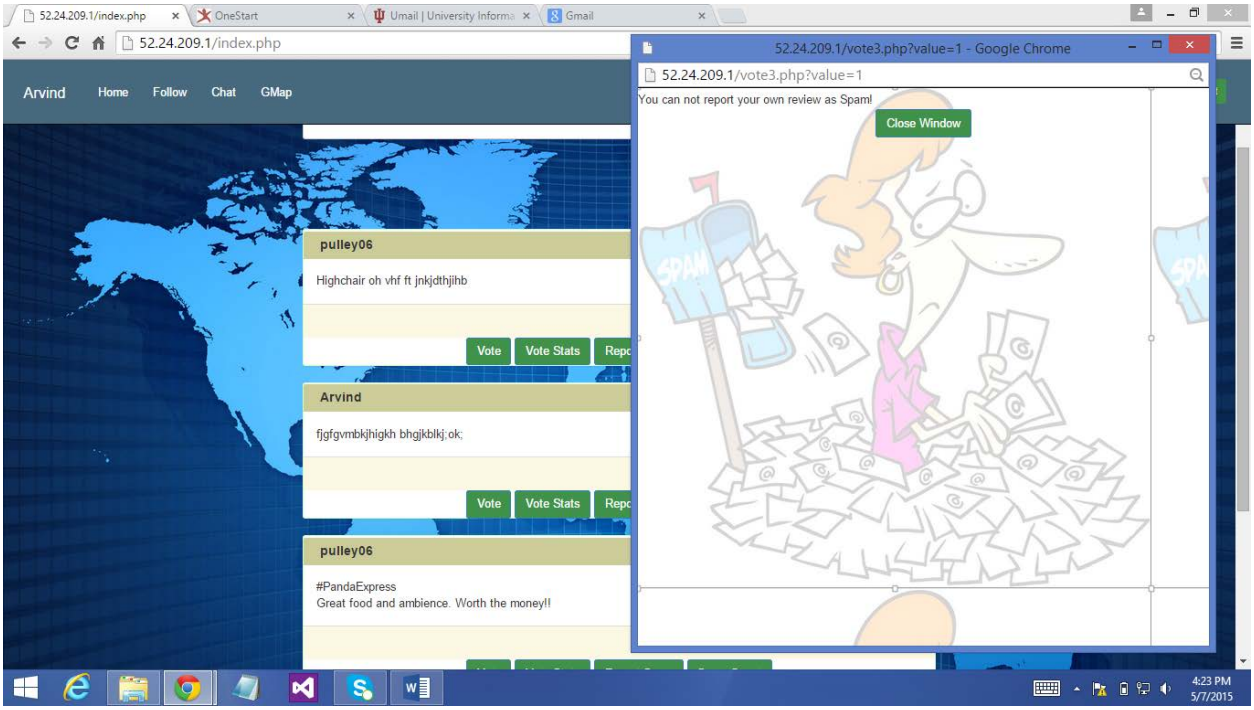


Figure 37: Error message when reporting own review as spam

In the screen shot above an error message is displayed after a user tries to report their own review as spam. In the image on the left the same is shown on an Android device.

Figure 38: Error message when reporting own review as spam

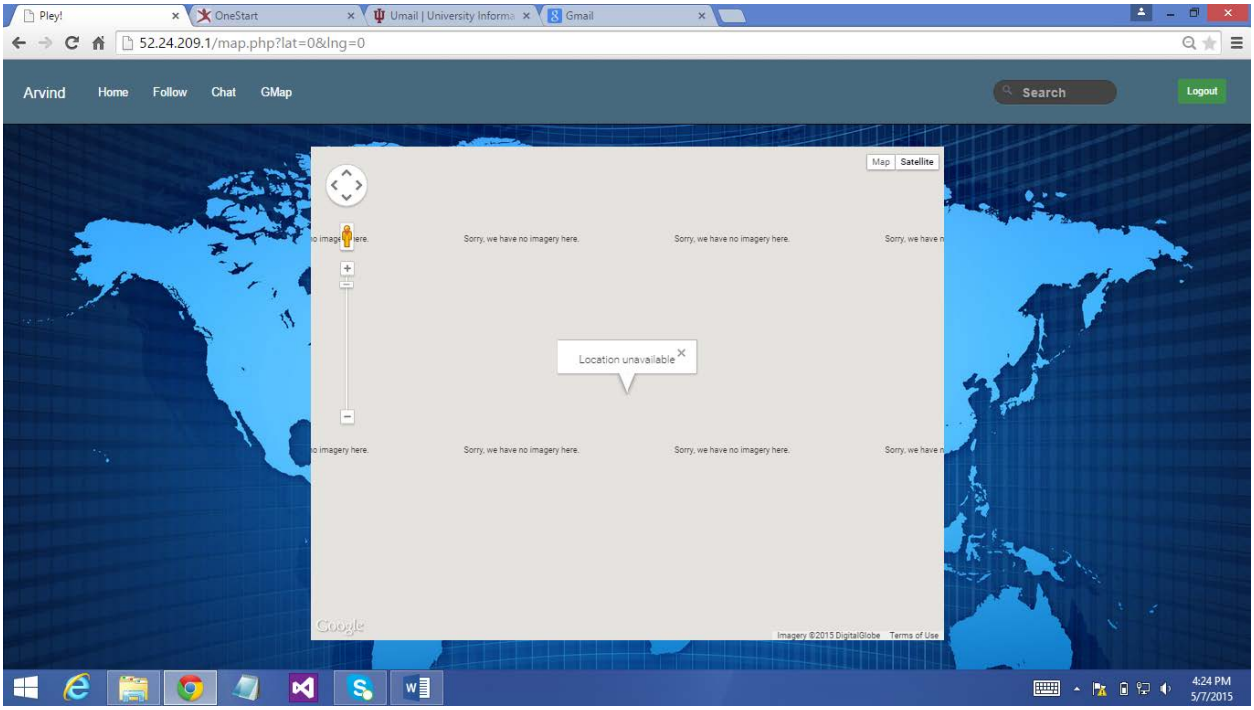
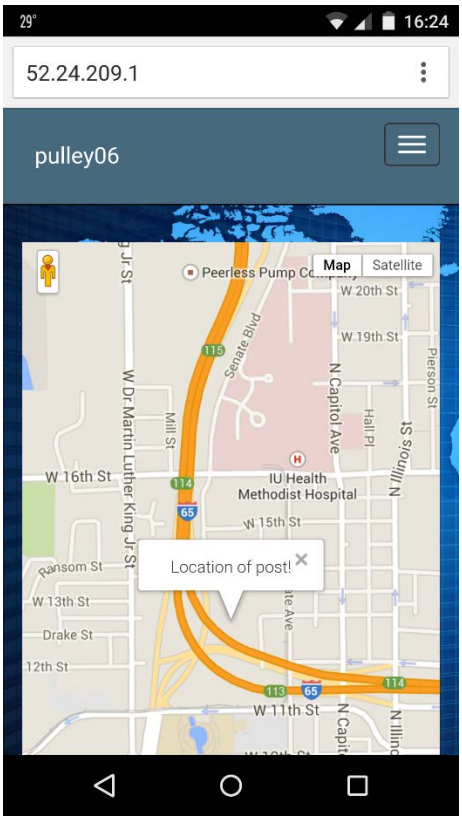


Figure 39: Review geo-location feature



In the image above, Arvind tries to see the location of the review as posted by user pulley06. Since pulley06 has not allowed for the location services to be used on his device, location unavailable is displayed.

In the image on the right, pulley06 checks the location of Arvind's review. This is available as Arvind had allowed location services on his PC.

Figure 40: Review geo-location feature

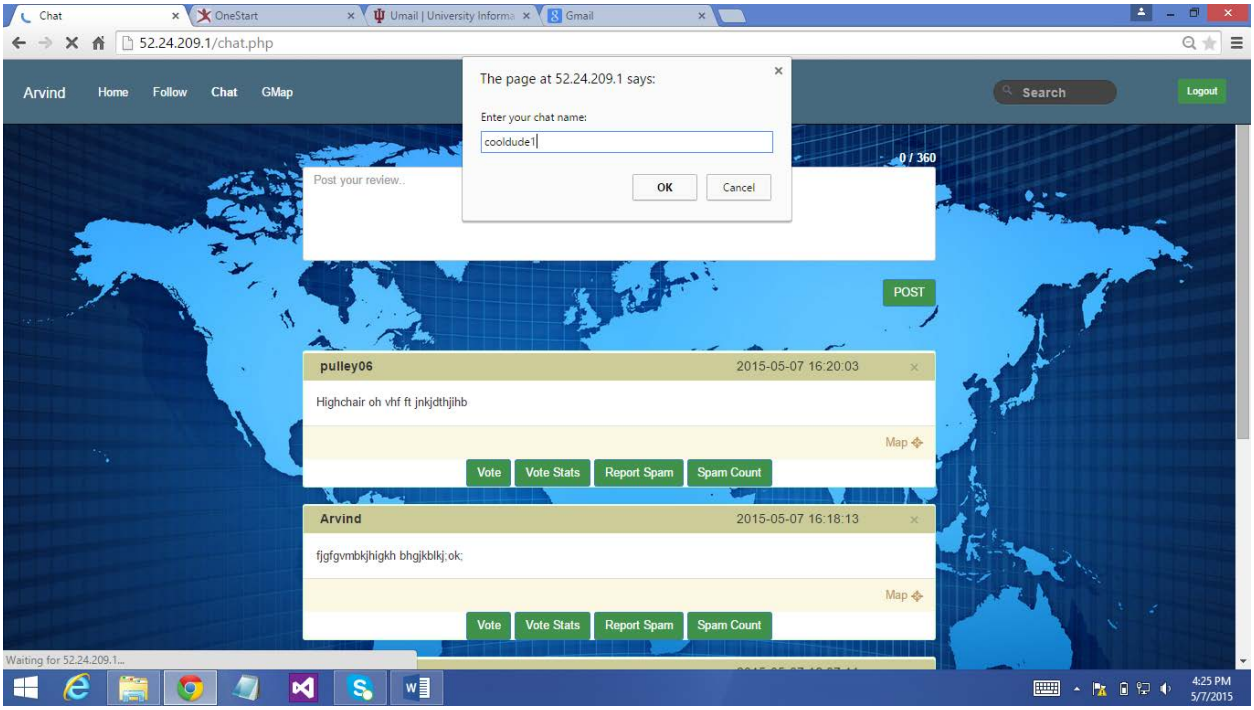
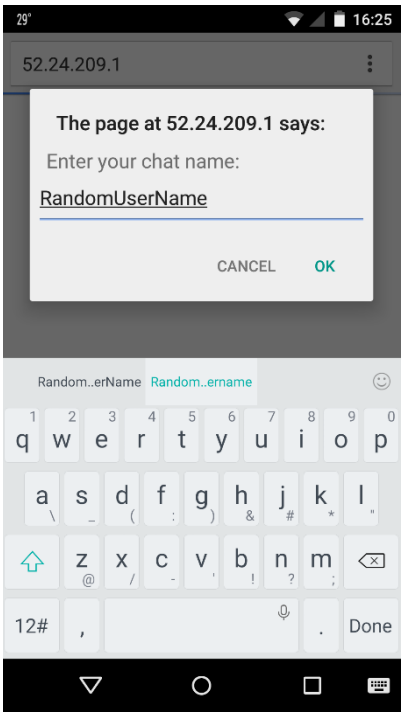


Figure 41: Prompt to add unique chat name



In the screen shot above the user has the opportunity to enter a chat name to access chat feature. The same is shown on the left in Android.

Figure 42: Prompt to add unique chat name

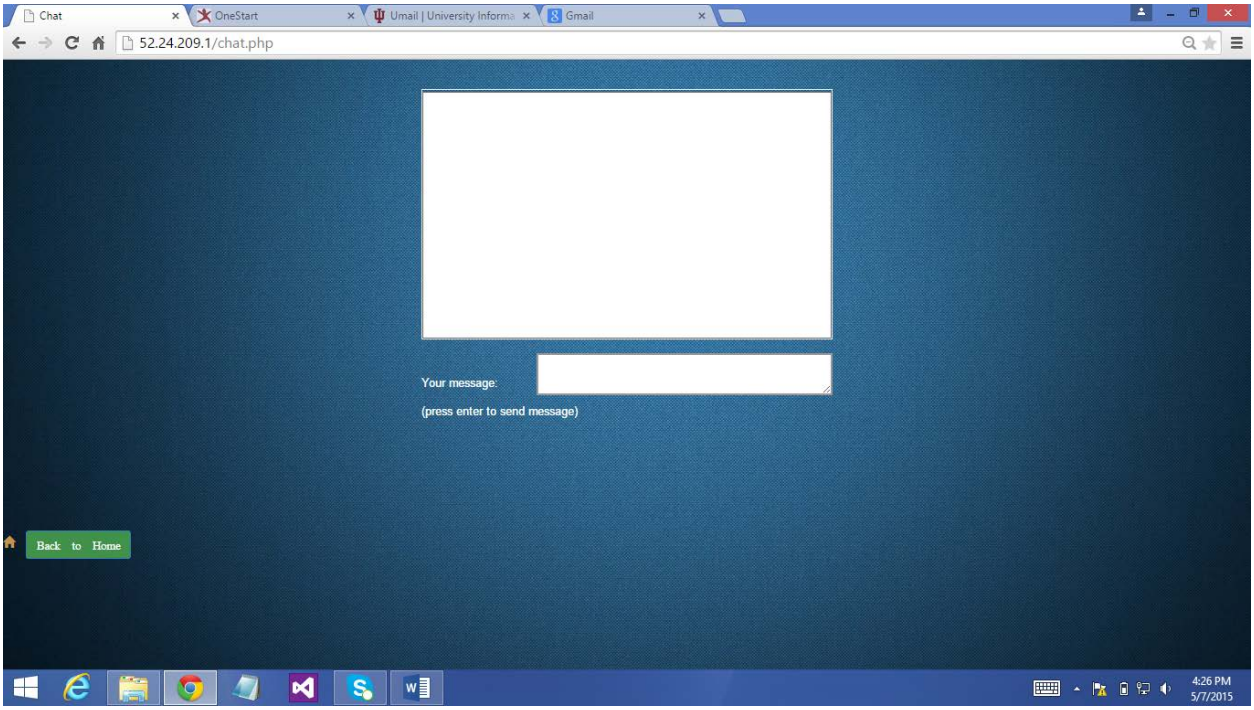
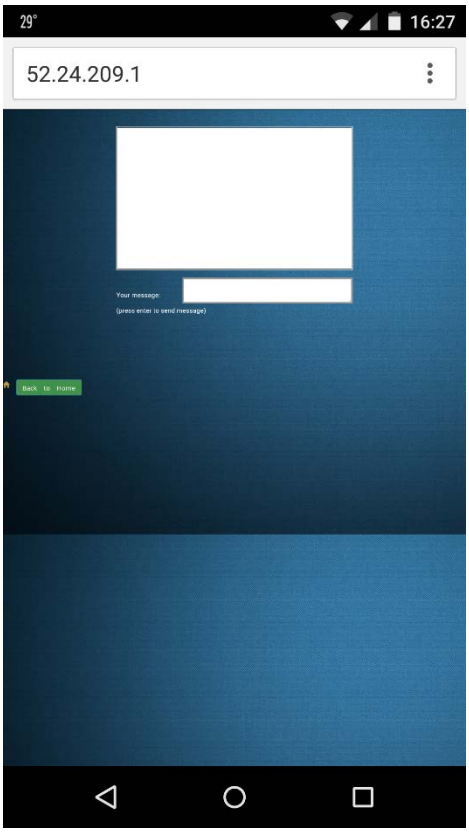


Figure 43: Blank chat view



In the image above a blank chat view is shown on the PC. The same is shown on the left on an Android device.

Figure 44: Blank chat view

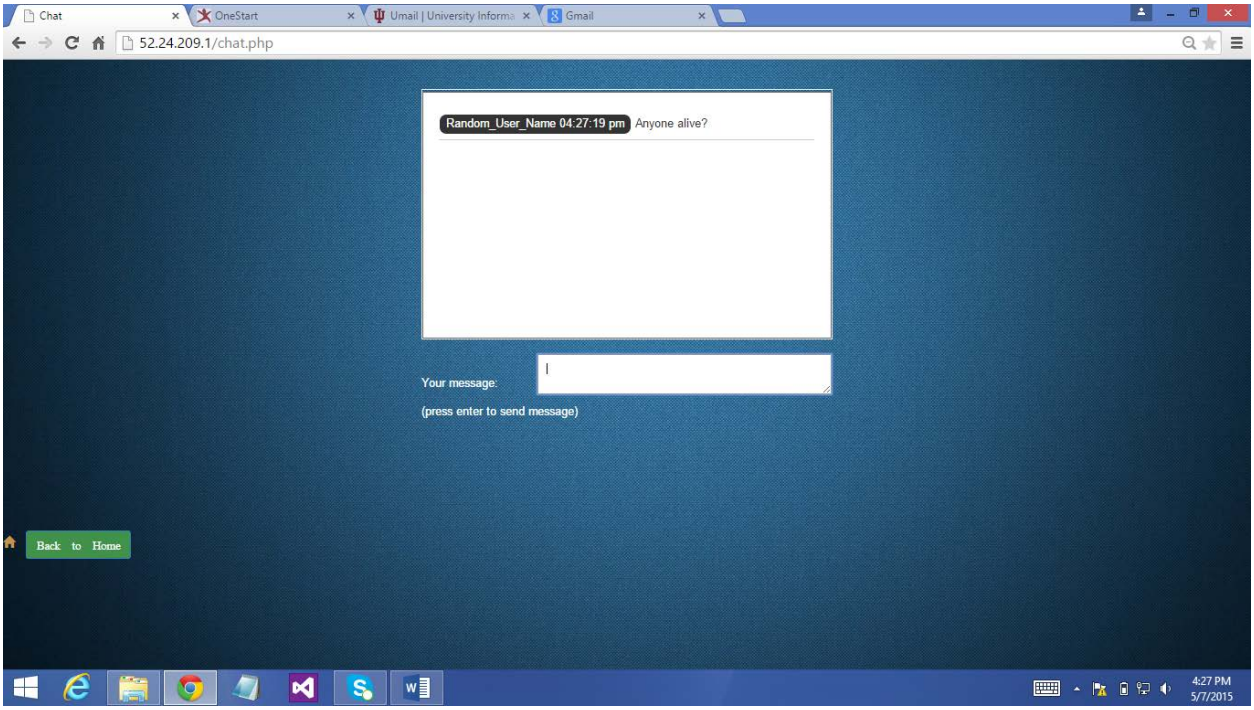
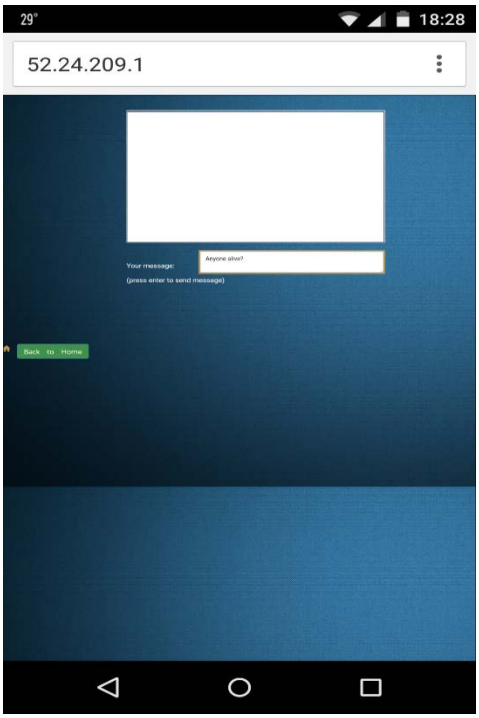


Figure 45: Chat image 1



In the image on the left user pulley06 with chat name Random_User_Name sends a message to every other user currently using the website and is in the chatroom. The screen shot above shows the message received from Random_User_Name with a time stamp.

Figure 46: Chat image 2

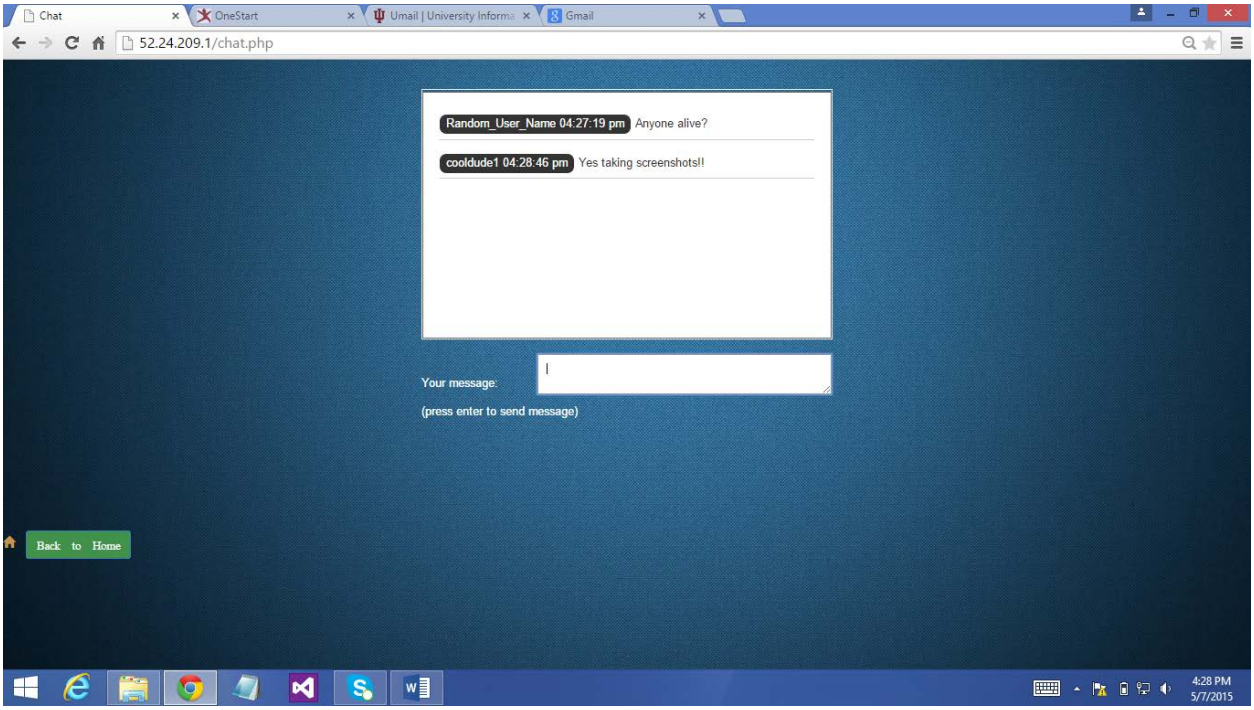
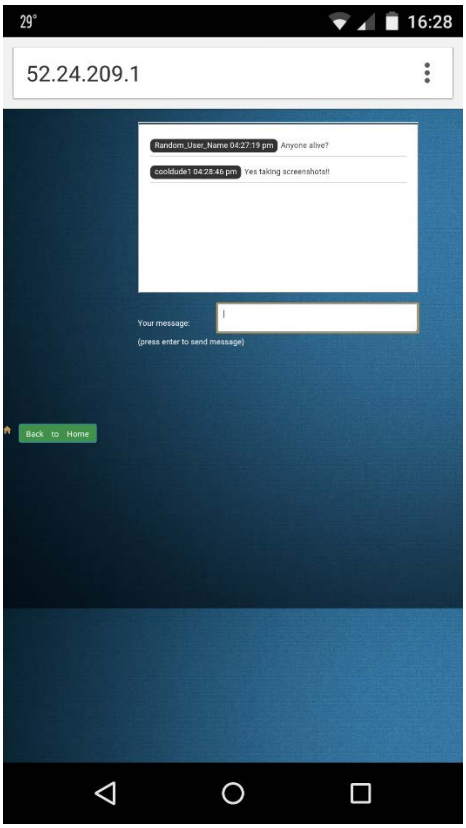


Figure 47: Chat image 3



User Arvind sends a message in response to the earlier received message. Pulley06 also receives the same as indicated by the screen shot on the left.

Figure 48: Chat image 4

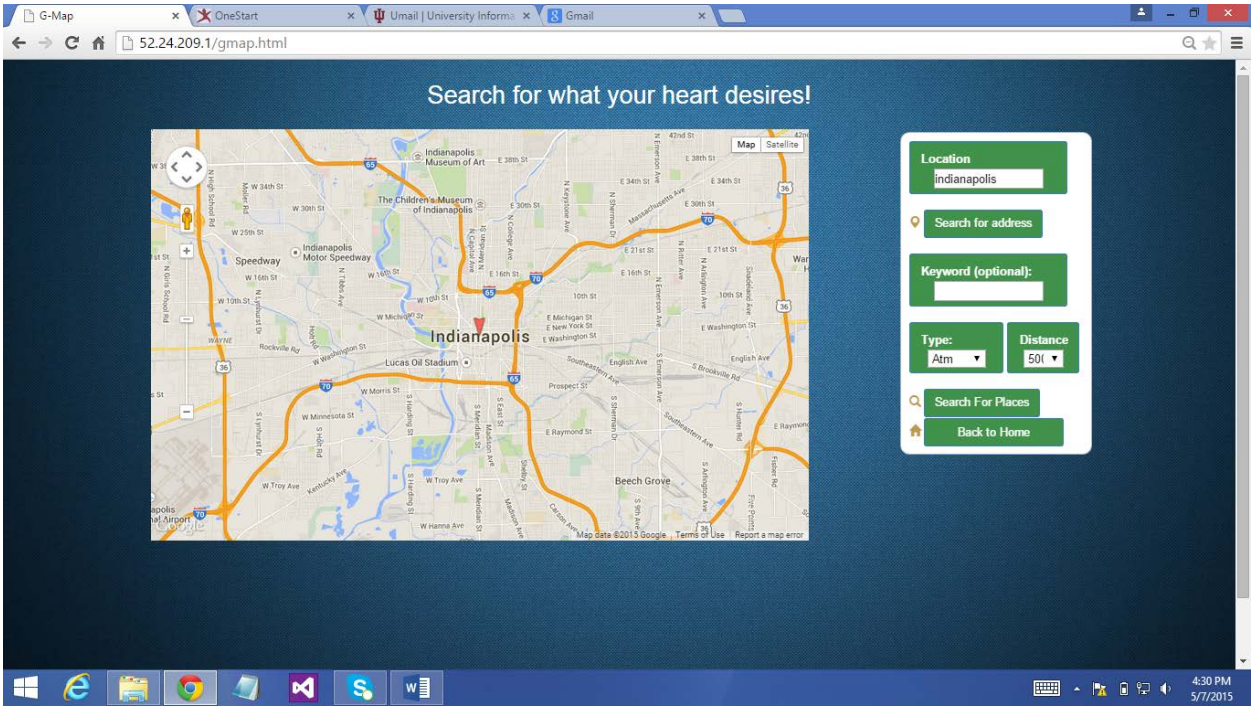
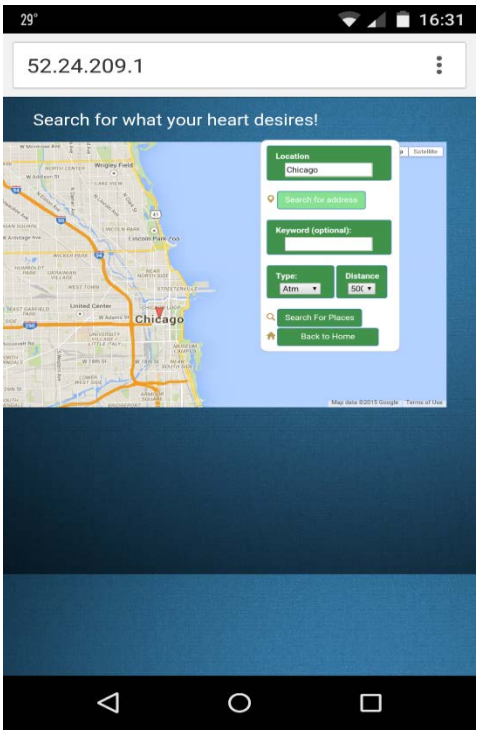


Figure 49: GMap on PC



The image on top shows Gmap on PC. User Arvind searches for ‘Indianapolis’. In the image on the left user pulley06 searches for ‘Chicago’

Figure 50: GMap on Android

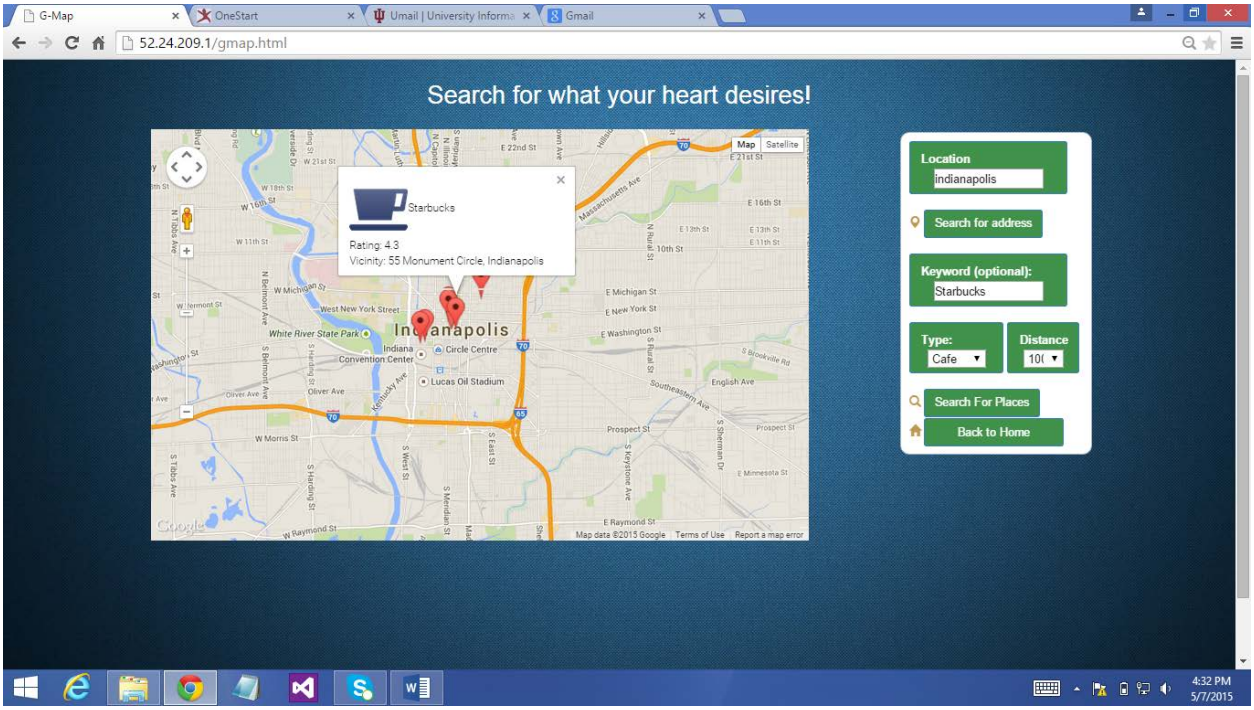
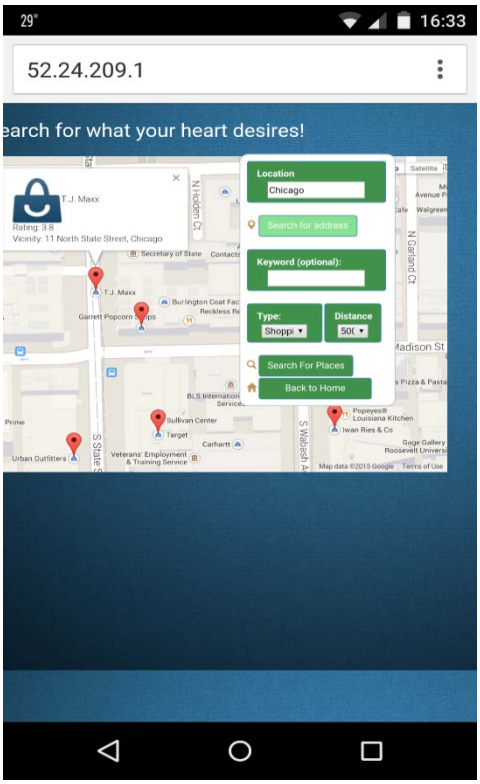


Figure 51: Arvind searches for Starbucks in Indianapolis with distance 1000



In the two screenshots the users can search for different places of interest.

Figure 52: pulley06 searches for shopping places with distance 500

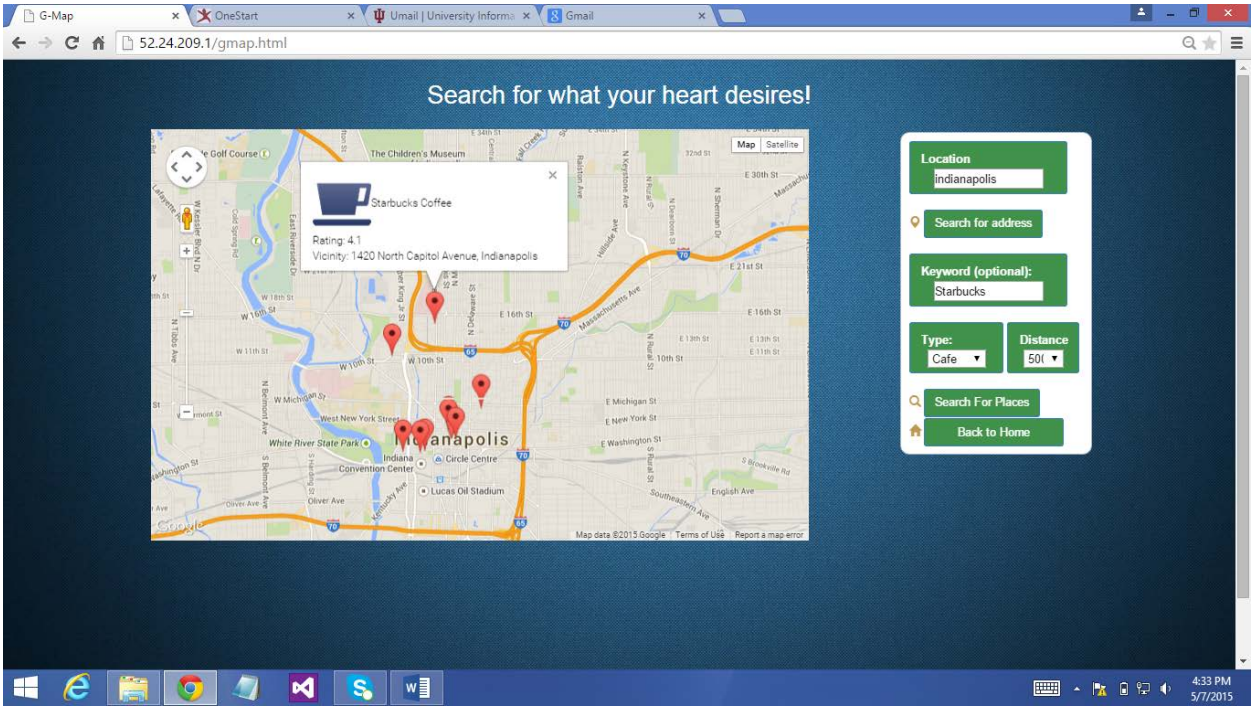
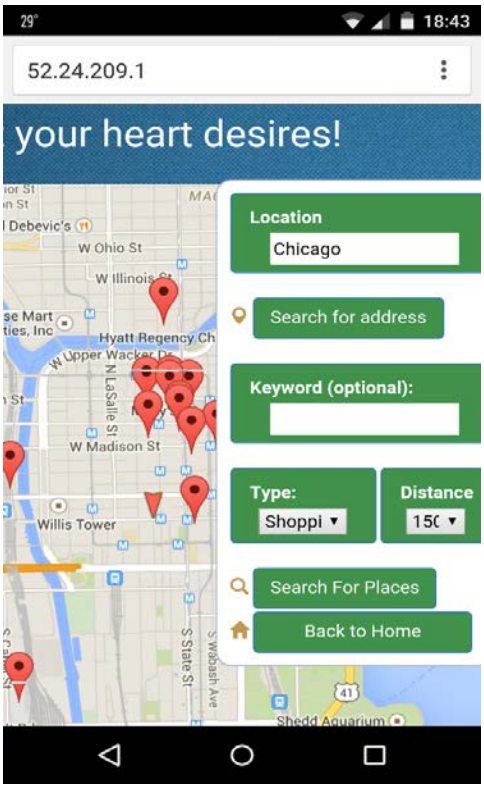


Figure 53: Same search with a larger distance



In the two screen shots shown here, the search distance has been increased to show more results in the search.

Figure 54: Same search with a larger distance

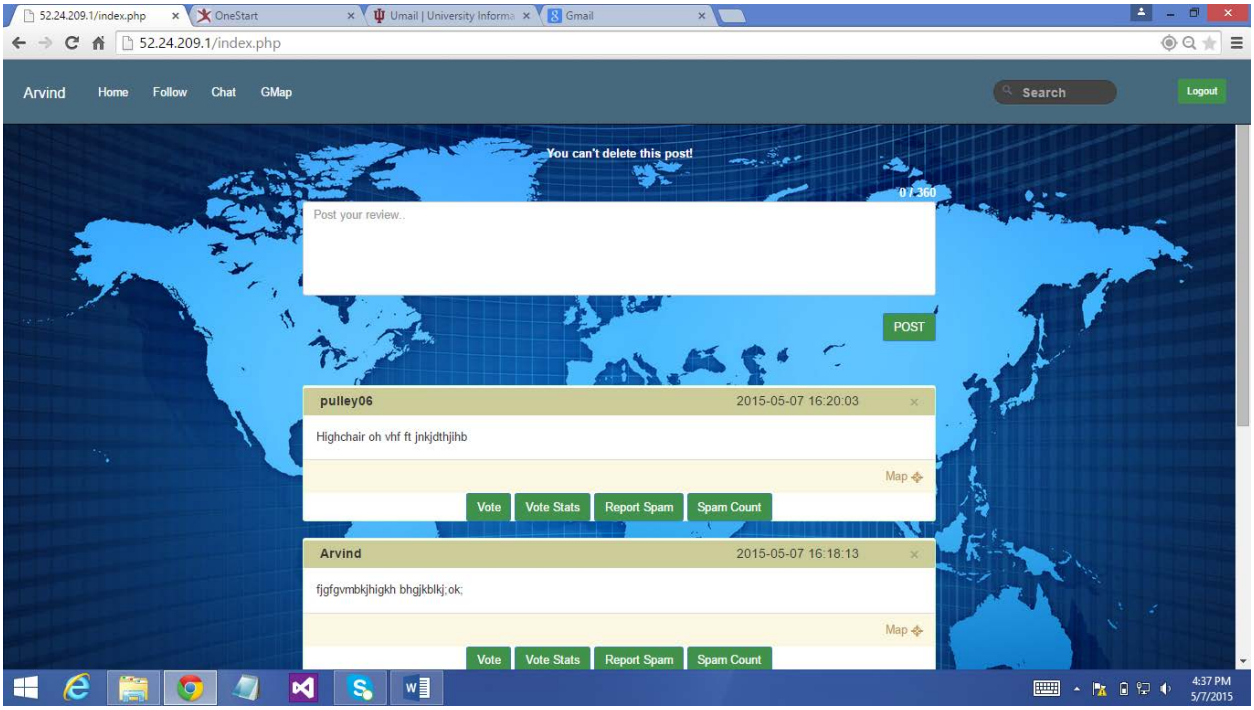
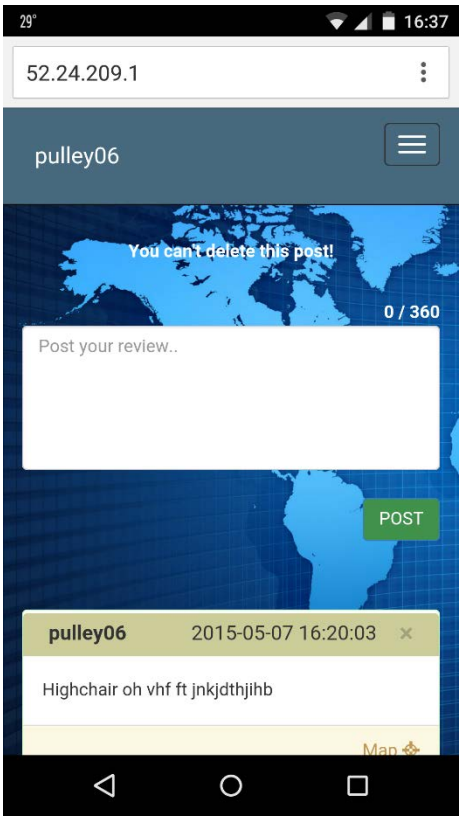


Figure 55: Error message when you delete a review not posted by you



The two screenshots on this page show the error message displayed when you try to delete a review that is not posted by you.

Figure 56: Same error message as Figure 55

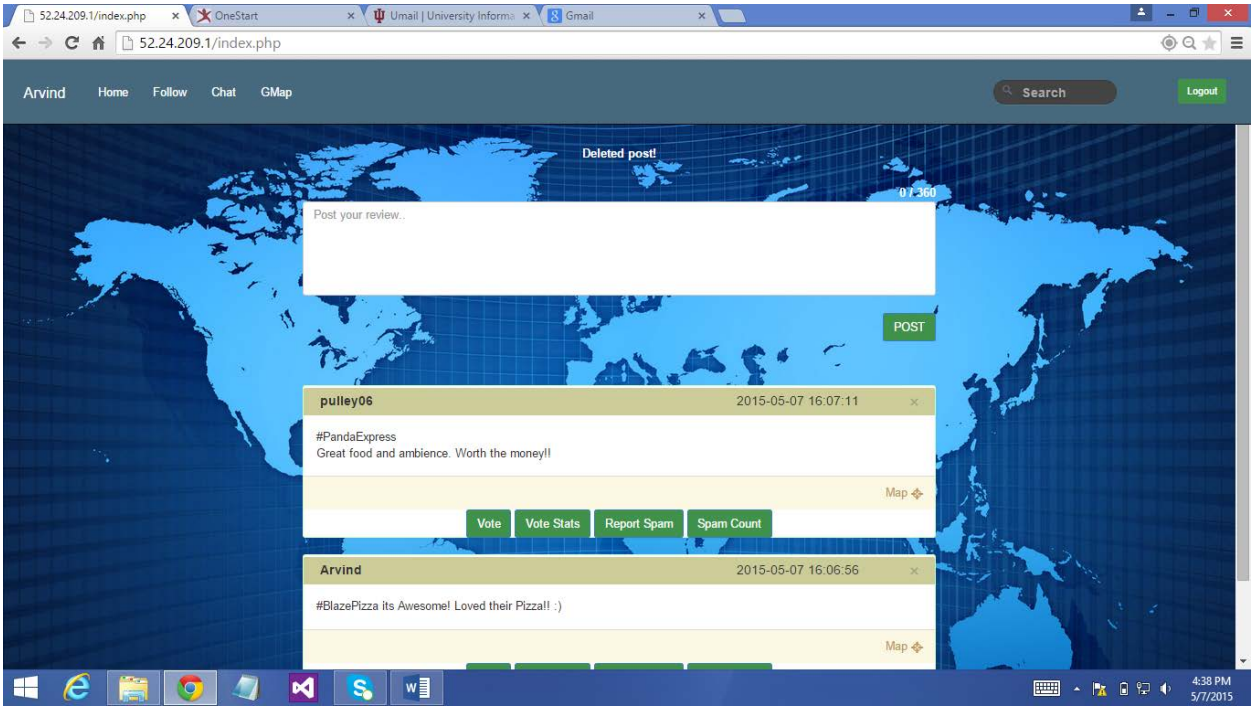
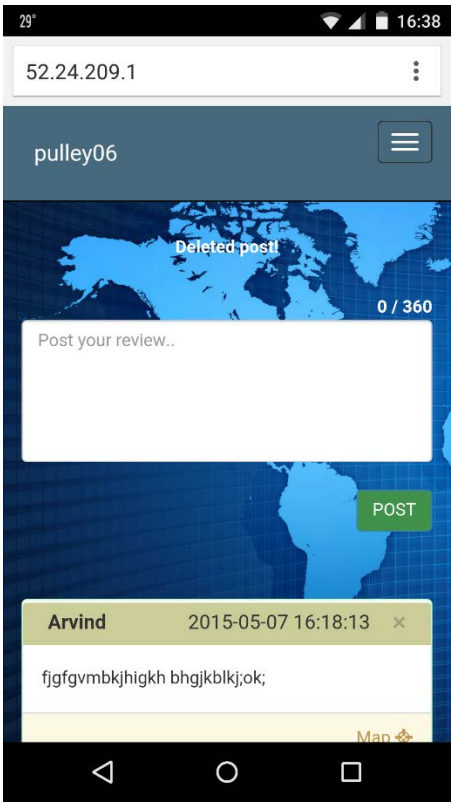


Figure 57: Deleted review successfully



The screenshots on this page show a message which is displayed when a review is deleted successfully.

Figure 58: Successfully deleted review

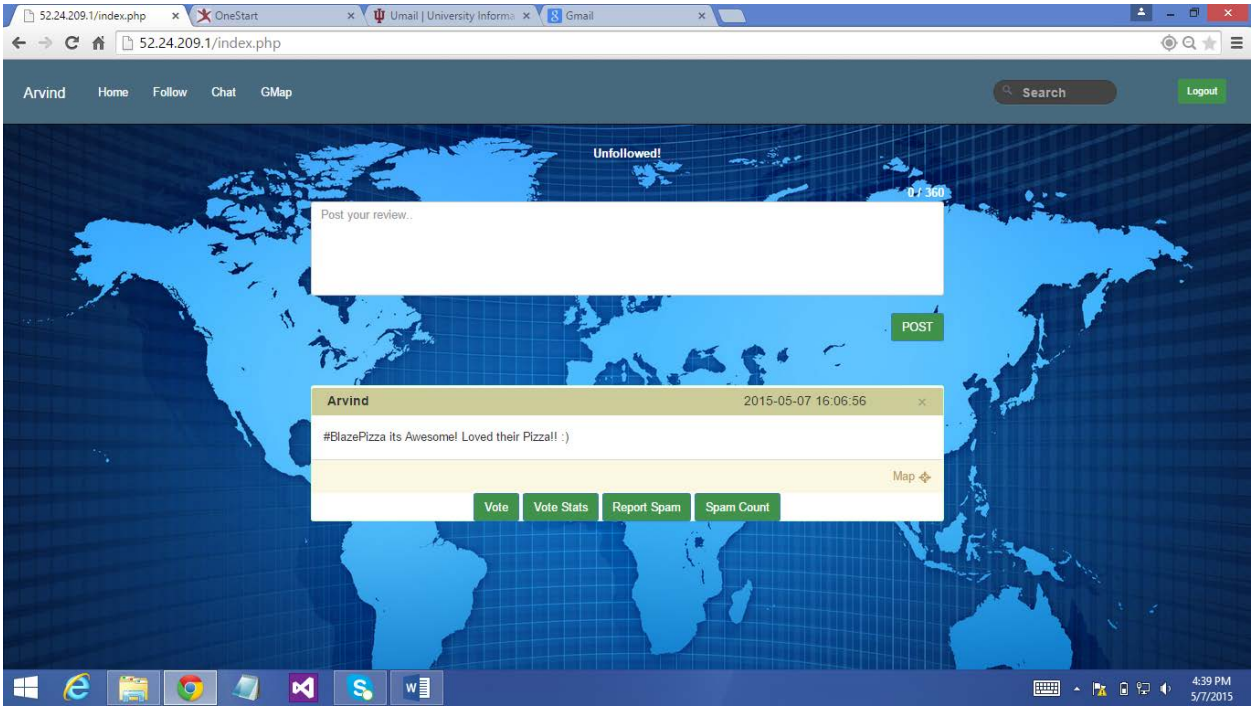
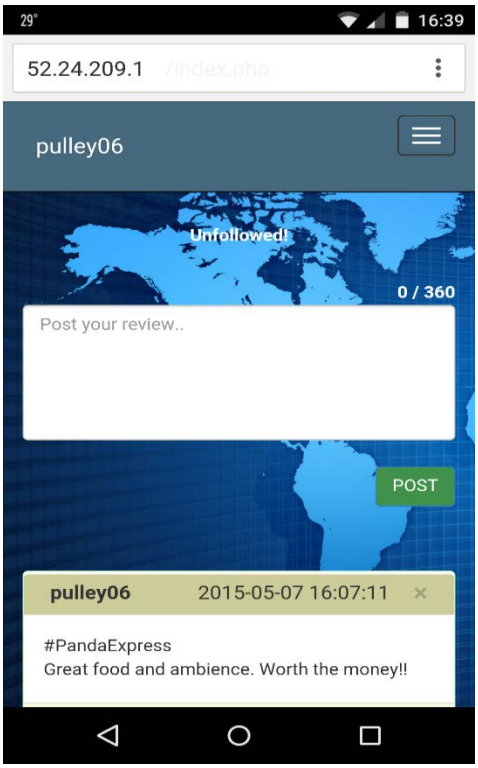


Figure 59: Message displayed after successfully unfollowing a user



The screenshots displayed on this page show the message when you successfully unfollow a user.

After this, you will not be able to see that user's post unless you follow her/him again.

Figure 60: Successfully unfollowed a user

11. Future Work

1. We can use Google recaptcha while signing in, login and reporting as spam, so that bots will not be able to post as users. It will effectively make only humans use the website and not bots or scripts. For this we need to register the website and obtain a domain name and use that for obtaining Google's reCaptcha services.

2. We can incorporate ads in our system. This would work as follows: Suppose a user a goes to the website and posts a review about B café in Indianapolis. The review will contain the geo-location on the map about where the review came from. This can be recorded in our system and based on this information we can use Data Mining as well as Recommender Systems concepts and display ads of cafes or restaurants in Indianapolis. The interested businesses would pay to have ads to display to people who are in that location/vicinity. So, Restaurant C in Indianapolis would pay the website to display its ad to User a prompting user A to visit C. The ads can also be specific to the review the user votes. If User A votes restaurant D in New York then we could display ads related to restaurants in New York similar to her/his interests.

3. We could also create a new page where users can select the location and get the top voted reviews for interesting spots in that location. This would work as follows: The user A would go to page topreviews and select a location e.g., Indianapolis. This would display the top spots as per user voted reviews. Here also we could have interested businesses place ads for showing their businesses on top as sponsored. The user A would also feel to visit those businesses. The businesses could maybe show some discounts for users which will prompt them to visit those spots. Thus, points 2 and 3 would serve as a good business model.

4. We could also show top reviews of a particular user's people, the user follows on a separate page. These reviews can be arranged as per descending order of number of votes secured. E.g., If User A follows users B, C, D, and E then She/he can show on the topfollowreviews page the top voted reviews of users B, C, D and E based on this week or this month or for a particular week or month or a particular week or month in a specific year. We could also filter by location. This would help the user identify trends and visit the places accordingly.

5. Although we have used Bootstrap for better UI experience on mobile devices, it is better to make a custom mobile app for this website. We could make separate apps for iPhone and Android users to use this website, fully customized as per their needs.

6. We could also use to propagate the reviews to different social media platforms like Twitter, Facebook etc.

7. To introduce the concept of verified/experienced users, a user who has less than 10 reviews must go through the Google reCaptcha before posting. Also, once a user has more than 10 posts with no spam report and an average rating of 3.5 and above on her/his reviews they can post without reCaptcha.

12. Contribution by Team Members

Contribution by Arvind Nair

1. Login Service
2. Index.php [home page] and its associated functions & connecting pages.
3. Search function to search for a particular review
4. Enabling map services on every review to view the geolocation of the post.

Contribution by Pulkit Sood

1. Registration service
2. Follow/Unfollow users
3. Ability to vote/report spam and check vote/spam statistics
4. Designing the database/queries

Contribution done as a team

1. Chat functionality
2. Gmap functionality
3. Deploying the project on cloud

13. References

1. Lynda.com
2. Google Places API - https://developers.google.com/places/
3. https://developers.google.com/places/webservice/
4. W3Schools.com
5. Amazon Web Services – EC2
6. css-tricks.com

14. Appendix

HTML – Hyper Text Markup Language
PHP – Hyper Text Preprocessor
CSS – Cascading Style Sheets
EC2 – Amazon Elastic Compute Cloud
API – Application Program Interface
AWS – Amazon Web Services
AMI – Amazon Machine Image
SSH – Secure Shell
TCP – Transport Control Protocol
HTTP(S) – (Secure) Hypertext Transfer Protocol
GUI – Graphical User Interface
UI – User Interface