NSS COLLEGE OF ENGINEERING

PALAKKAD, KERALA – 678008

UNIVERSITY OF CALICUT

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MINI PROJECT REPORT

2014-2018

CAMBUZZ

SUBMITTED BY

ANANDHU K K	**NSAOECS008**
ANISH UNNIKRISHNAN	**NSAOECS011**
GOKUL P V	**NSAOECS024**
JOYAL JOSE	**NSAOECS032**
KANNAN K	**NSAOECS033**

GUIDED BY

Mrs. Remya G

Assistant Professor

Dept. of Computer Science & Engineering

N.S.S.C.E, Palakkad

NSS COLLEGE OF ENGINEERING

PALAKKAD, KERALA – 678008

UNIVERSITY OF CALICUT

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that this is the bonafide report of the mini project entitled“ **CAMBUZZ ”** done by **ANANDHU KK, ANISH UNNIKRISHNAN, GOKUL PV, JOYAL JOSE** and **KANNAN K** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer science & engineering under theUniversity of Calicut.

GUIDED BY

Asst. Prof. Remya G

HEAD OF THE DEPARTMENT

Dr. Prof. Viji Rajendran

STAFF IN CHARGE

Asst. Prof. Kiran V K

ACKNOWLEDGEMENT

Many noble hearts contributed immense inspiration and technical assistance for the successful completion of our mini-project work. We are unable to express our gratitude in words to such individuals.

First of all, we thank the Almighty **God**, for granting us the strength, courage, and knowledge to complete this project successfully.

We acknowledge our sincere thanks to the management of N.S.S College of Engineering for their help in the successful completion of our project. We would also like to express our deep regard to our principal **Dr.Prof. Sudha T**, for making our mini-project a grant success.

We take this opportunity to express our profound gratitude to our project guide and Head of the Department, **Dr. Prof. Viji Rajendran**, for all her guidance and valuable support towards making our project a grant success.

We are also thankful to our teachers, **Mr. Kiran V K, Mrs. Remya G, Mrs. Sruthy Manmadhan** and **Mrs. Maanasa N A S** for their valuable guidance throughout our project.

We would also like to thank all the non-teaching staff of our department for their constant encouragement throughout our project. This helped us in proper completion, installation and demonstration of our project. Last, but not the least, we take pleasant privilege in expressing our heart full thanks to our friends who were of precious help in completing this project.

ABSTRACT

In recent years, social networking system hasbecome one of the most popular Internet applications and hasa large number of users. Although the public social networkingsystems have met the interpersonal communication demand of people, there is inadequate support for the closecombination of user's learning, working and cultural life on campus with their real activity.Any technology that is able to captive so many students for so much time not only carries implication for how that student view the world but also offers an opportunity for educators to understand the elements of social networking that students find so compelling and to incorporate those elements into teaching and learning.

This project aims to reshape campus communication and community through social networking site.The project design and realized a new campus social networking system for users, which is tightly integrated with the reality environment of campus and users' real activity. Both the students as well as the faculty can use this network at the same time. It will not only achieve the basic communication function but also provide a unified platform for the teaching, researching, management and many other aspects of cultural life to teachers and students oncampus. On the purely social level, they offer students avast amount of information about one another and when used as an organizing tool, they provide new ways for students group to reach out to members, share information, learn about campus events, and mobilize support and action.

CONTENTS

1 INTRODUCTION

Social Networking - It's the way the 21st century communicates now. Social networking is the grouping of individuals into specific groups, like small rural communities or a neighborhood subdivision. Although social networking is possible in person, especially in the workplace, universities, and high schools, it is most popular online. A social network site is defined as web-based service that allows individuals to

- Construct a profile within a bounded system
- Articulate a list of other users with whom they share a connection
- View and traverse their list of connections and those made by other within the system.
- Can share photos, music and blog posts, and videos with one another and can comment on content posted by other peers.

The nature and nomenclature of these connections may vary from site to site. Since their introduction, social network sites (SNSs) such as Facebook, twitter have attracted millions of users, many of whom have integrated these sites into their daily practices.

In the last few years, social network sites (SNSs) such as Facebook have become standard fixtures on college campuses and they now constitute an integral part of the daily communication practices for many students. For those of us involved in higher education, it is important to understand SNS development, practices, outcomes, and motivations for use because these sites are fundamentally changing the social fabric of colleges and thus a wide range of college activities. On the purely social level, they offer students a vast amount of information about one another. When used as an organizing tool, they provide new ways for student groups to reach to members, share information, learn about campus events and mobilize support and action.

2 REQUIREMENT ANALYSIS

2.1 SCOPE OF THE PROJECT

The primary idea for developing this project is to provide an online platform that is intended to facilitate direct communication between staffs and students of our college.

Although public social network services like Facebook, Twitter etc. provide good individual and group communication services to their users, they are isolated from campus cyberspace of campus users. It cannot support the learning, teaching, research and cultural activities on campus closely and timely; on the other hand, public social network systems cannot provide strict information and privacy protection to their users and organizations that users belong to because of their consideration of commercial interests. Due to lack of effective coordination of these communication channels, communication in campus is ineffective. In colleges, the system for sharing information between students and systems has been conventionally the notice board and word of mouth. Even though they stay relevant, they are not always helpful at times, when a delay in passing information is undesirable and effective coordination of communication channels are required. It is in this context, that our project plays a crucial role.

Our project, Cambuzz introduces an effective way to facilitate and encourage knowledge sharing and knowledge creation on campus which eliminates the problems like delay, security, privacy and lack of common platforms. Through this system, we believe that this can make a drastic change in knowledge sharing and enhance the communication between the students and the campus authorities.

2.2 IDENTIFIED SYSTEM REQUIREMENTS

2.2.1 USER DETAILS

There are two main categories of users

- Authorities
- Casual users

Casual Users: A casual user can be a faculty or a student. The main features provided for the casual end users are

- They can update their profile information.
- Can post on the wall and also like/comment on the posts.
- Can view news and announcements made by the campus authorities in the campus dynamic.
- Can involve in discussions in the forums.

Authorities: Just like a casual user, an authority has a profile and a wall. And they are authorized to post on campus dynamic. The main authorities are the principal, one representative from each department and representatives from union, cells, and clubs. They are given a prior username and password,in which password can be changed conveniently.

2.2.2 HARDWARE REQUIREMENTS

- **Processor**: Intel Pentium IV 2.0GHz and above
- **RAM:**512 MB and above
- **Hard disk:**80 GB and above
- **Monitor:**CRT or LCD monitor
- **Keyboard**: Normal or multimedia
- **Mouse:** Compatible mouse

2.2.3 SOFTWARE REQUIREMENTS

- **Front End :** HTML,CSS, JS
- **Back-End:**PHP
- **Local Server Package:**WAMP Server
- **OS:**Windows XP or above
- **Web Browser:** Any latest browser

2.3 PRIORITIZED REQUIREMENTS

The requirements are listed out in the order of their priority:

- **Database:**The main database containing information regarding the users should be maintained and everything a user post on their wall or on a common page including text, images should be updated. The data generated by likes and comments should be properly maintained in the database on their respective tables.
- **Server:**The Server acts as a wrapper for all server functions of our social networking web application. It essentially acts as a link between all of the information such as accounts, account details, posts, likes etc. to the main database.

2.4 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

Functional Requirements are:

- **Account Creation**: When a user accesses the web app for the first time, he/she must create an account before using any of its features. The signup form mainly consisting of basic information about the user including the name. All the fields are mandatory. The purpose of the password and email id confirmation is to ensure that the user didn’t accidently mistype these details. User clicks signup, a new account and account details are added to the database through the server, and the user is brought back to the login page.
- **Account:** Each user who wants to use the site must create an account. This is the main module use to determine what a user does and when the user did it. The account's

information has two put poses: hold profile information, and hold privacy information with such a large amount of information to keep track of. The Account module would be very large and difficult to work with. Therefore to ease the load, the Account module was broken up into two different classes. There is the actual account class which keeps track of profile information. And the other module is the account settings, in order to update the user details. There are two cases in which the profile module needs to be updated, one is when the user changes his/her username and/or password and the other is when auser posts something on the wall.

- **Account Settings:** An Account setting is a module within the profile module. The page meant for modifying user's profile. Some limited information including the profile picture, skills, and the password can be changed using this module.
- **Posts :**As people use their social networking account, they will want to be able to upload interesting images, texts to share with their friends. They are collectively called as a post. The wall, dynamic, and forums are the pages intended for posting. Each of these pages contains a post upload area in which the user can enter the texts or select a picture and then click upload which updates their corresponding tables in the database and is updated on the respective pages. The allowed extensions for images cover popular image extensions such as png. Jpg, gif, and bitmap. The inner workings of this uploading process are surprising. After the user clicks upload, the file is sent to the server and extension is then checked against an array of allowed extensions. If the extension is on the list, the file is then moved to a permanent location in the media folder under a subfolder for the page in a random name. Through PHP all information about these files can be accessed through variables. The location is stored in that variable and can be used to pull the file name and file extension. These posts contain options for like and comment which works like any other social networking site.

Non-Functional Requirements

- **Secure access** to confidential data by username and password. This is super secure for all kind of its users. Only the account owner can edit or post his own ideas and suggestions by our secure access.
- **Better component design** to get better performance at peak time.
- **Robust, reliable** and **fast** database for quick access and fast response.
- **Scalability:** The current system only consists of basic functionality meant for only current college students and college authorities. It can be extended to include alumni, retd. Teachers with the functionality of Messaging, Video conferencing etc.
- **Availability:** The system will be available for 24/7.

3 FEASIBILITY ANALYSIS

3.1 OPERATIONAL FEASIBILITY

The system will create a web-based interface where authorities and students can easily share information and users can view and review, it eventually improves the knowledge sharing and knowledge creation on campus. Through campus dynamics, campus authorities can share any information regarding news, event announcements etc. Forums basically provide an open page for discussion about anything and any valid user can view, comment on it. The wall provides a basic page for users to post their texts or images on the common page. So it provides abest efficient method for sharing and creating informationoncampus. Since it can be accessed through web browsers, it is easy for any student to access the system using any mobile phone or a PC. So the project is operationally feasible.

3.2 CULTURAL (OR POLITICAL) FEASIBILITY

The web application is developed in an environmental and cultural friendly manner. The Cambuzz is beneficial to the students on campus as well as for campus authorities. The project does not lead to usage of any kind of equipment or gadgets which are harmful to the society and is done in an ethical way to bring it to reality hence making it culturally feasible.

3.3 TECHNICAL FEASIBILITY

To develop the application, a high-speed internet connection, a database server, a web server and suitable software development kits are required. The internet is easily available everywhere. All teachers and students have access to the internet. Also, all of them possess a mobile phone, many of them being smart phones mostly android that have the capabilities to access the internet on the go through any browsers. Thus targeting the internet as a tool and these web browsers as a platform, the system improves the information sharing in between students and campus authorities. The software used for the development like HTML, CSS, JAVASCRIPT and wamp server is all the free software that developers have easy access to. Hence, it proves that the proposed system is technically feasible.

3.4 SCHEDULE FEASIBILITY

The Schedule for the project is as follows:

Stage 1 (Jan 30 - Jan 6): Topic Selection

Stage 2 (Feb 7 - Feb 13): Feasibility study and Requirement analysis

Stage 3 (Feb 14 - Feb 20):Web interface development

Stage 4 (Feb 21 - Feb 27): Initial integration of the database with the web interface and system demo (I)

Stage 5 (Feb 28 - Mar 6): Implementation document and testing document

Stage 6 (Mar 7 - Mar 13): System Demo (II)

Stage 7 (Mar 14 - Mar 20): completing the final features and final Testing

Stage 8 (Mar 21 - Mar 27): System Deployment

The current plan of activities suggests that this project is feasible with the current schedule constraints.

.

3.4 ECONOMIC FEASIBILITY

The project relies heavily on Free and Open Source Software. This helps to develop the project with theminimal cost incurred. The language and editors can be used without any cost. The sublime text editor, web browsers, all other languages used and database management software are free software which can be readily downloaded from the internet free of cost. During the design phase, the web application will be run on a web browser which is a part of any operating system. All other factors in this project are absolutely free of cost, and hence, so as far as the functions and constraints concerned, this project is economically feasible.

3.6 LEGAL FEASIBILITY

While operating the service, it should be monitored that the information and design do not violate any existing copyrights. Users will be asked to respect national and international copyright laws. Since the source code of the system will be licensed using a very permissive free software license (such as GPL or MIT), using free and open source software for development are not expected to pose any legal difficulties. Also, all the development tools and software to be used are freeware which can be downloaded from the internet without violating any piracy laws, thus declaring the project to be legally feasible in all means.

4 SYSTEM DESIGN

4.1 DATA FLOW DIAGRAM

It is a two-dimensional diagram that explains how data is processed and transferred in as a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output.

LEVEL 0:

The context diagram of the system is shown in figure 3.1.1.Level0 DFD orthecontextdiagram is the highest level view of the system and shows the only the external entities, casual users, and authorities. The system is represented as a single module which comprises of all sub-modules of the complete system.The Cambuzz represents the system as a whole. And a database which is used to store data generated by the system.

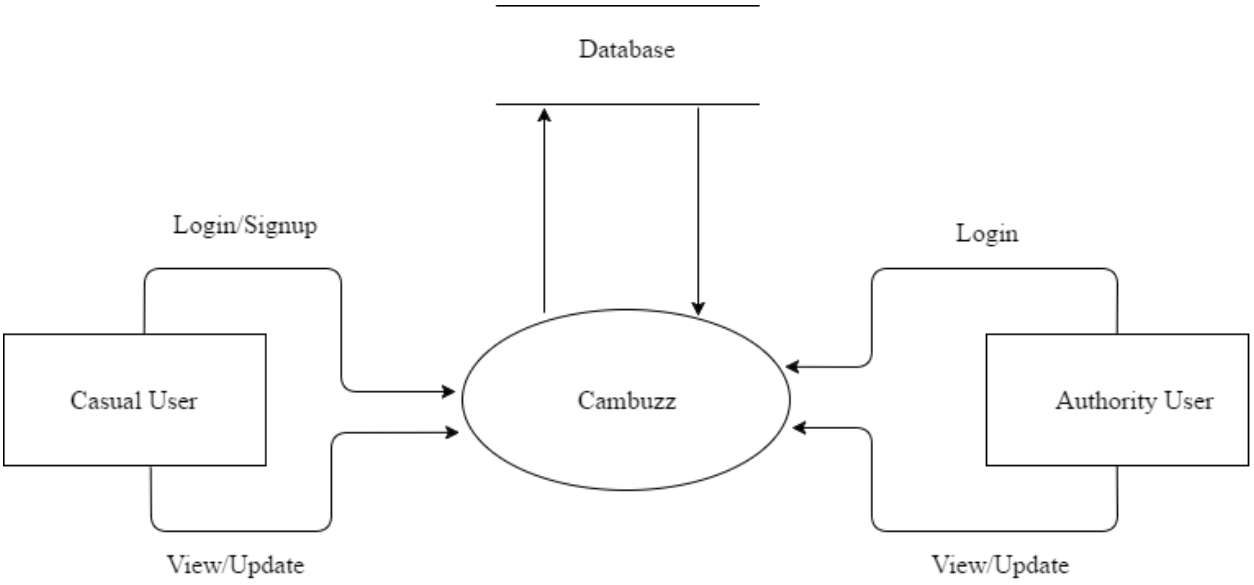


Figure 4.1.1

LEVEL 1:

The level 1 DFD of the system is shown in figure 3.1.2. The level 1 DFD is constructed by identifying the major components of the system. Here the main components are Registration, Login, Wall, Forums, Dynamics, Search, Profile. These components are updated by a database. The user login after successful registration. After the successful user will be directed to home/wall page. And from that page user can visit forums, campus dynamics, profile. A search option is also provided, which enables the user to view profiles of any other valid user.

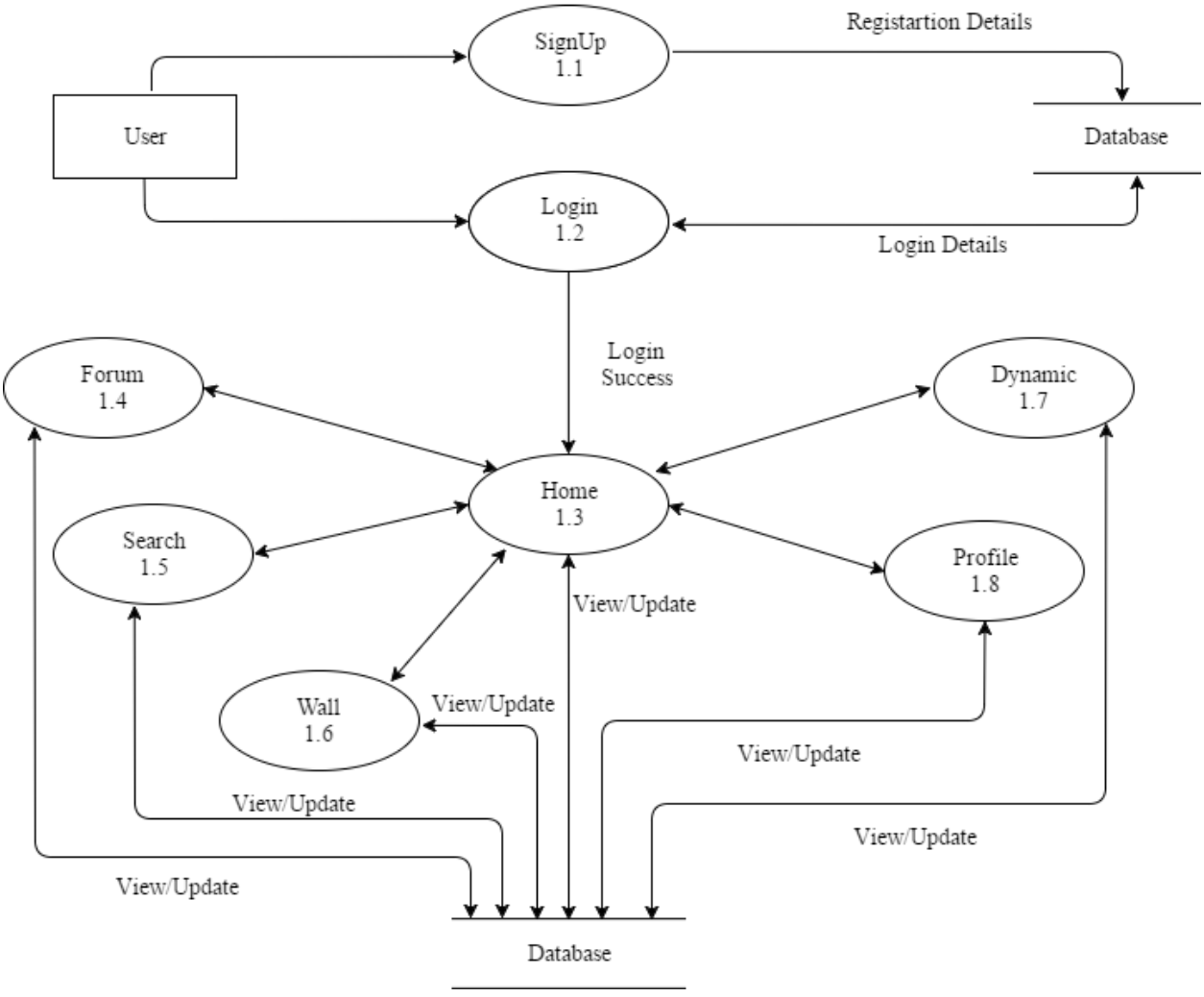


Figure 4.1.2

LEVEL 2:

The level 2 DFDdescribes the modules defined in level 1. There are seven components can be described in level 2 DFD. They are

Registration :

In the registrationprocess,the users are asked to provide somerequired details like name, e-mail id, password, and other personal information. Once the user entered the details they are validated using the current student details of the college. The validated details are then updated to a database. And the database returns a validation status, which confirms the registration of the user. Once the student/staff completes the registration they can login using the name/e-mail id and password they provided during the registration phase.

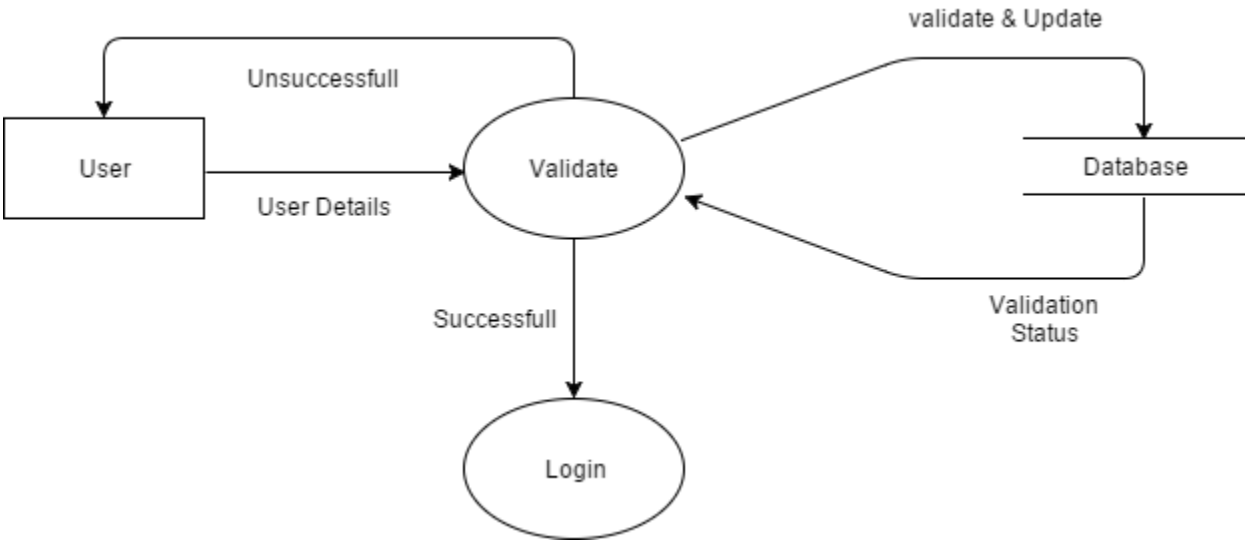


Figure 4.1.3.1

Login :

The user has to input email id and password of their account which then validated using the current information stored in the database. If the details are correct the user will be directed to his/her home/ wall.

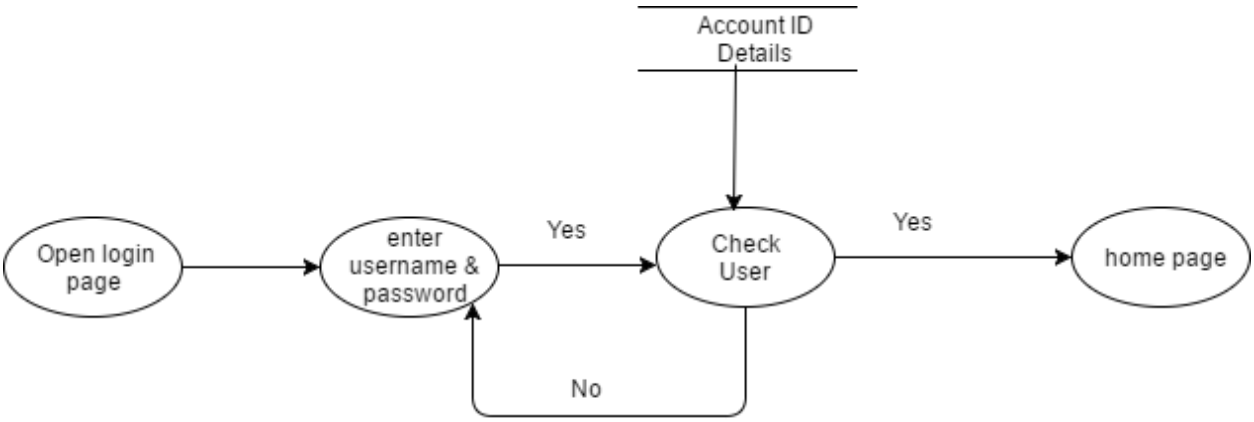


Figure 4.1.3.2

Home:

After the successfullogin, the user will be redirected to the home page. Home pagecontains information about the cambuzz and about the college. For the authority users, the page is intended for posting on to the dynamic page. The posts uploaded form the home page will be updated in the database and as well as in the respective table on the database.

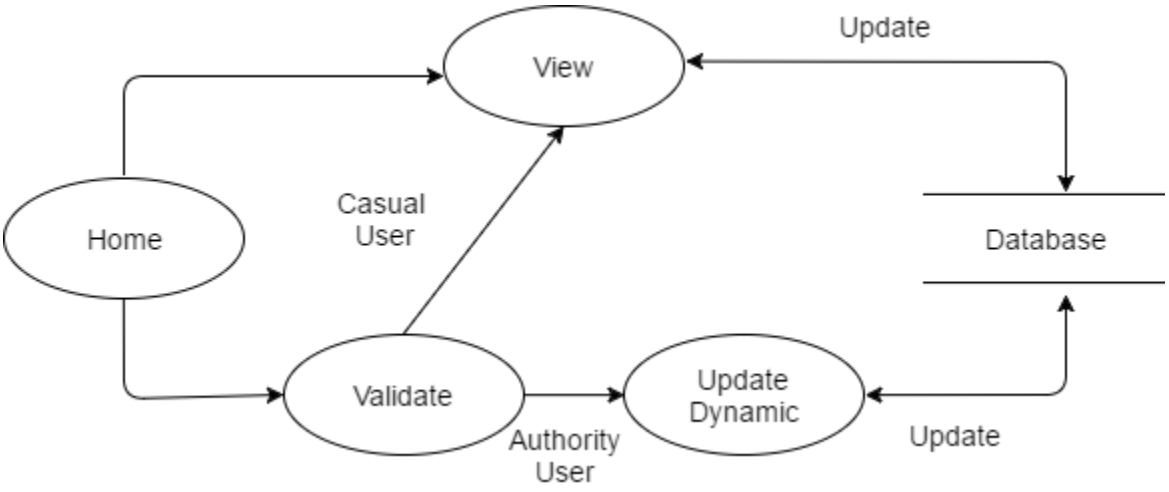


Figure 4.1.3.3

Wall:

Wall contains the posts that were posted by users. The wall consists of posts posted by users from the profile page. Viewing the wall comprised of like, comment on any posts. The updatecan

be adding or removing posts. Adding phase requires theuser to enter the text or select the image the user wish to add on the wall. These all are updated to a central database.

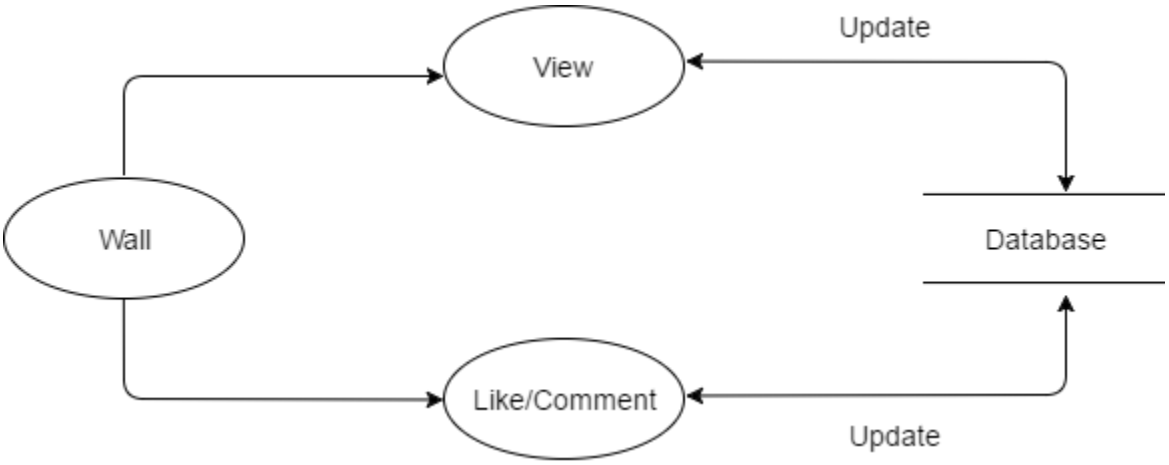


Figure 4.1.3.4

Campus Dynamics:

A campus dynamics page is intended to provide details about the activities undertaken by different campus authorities like office, union etc. The casual users are only allowed to view the page. The campus authorities are authorized to edit the page. Once the user enters the campus dynamic page they are authorized to check whether they belong to casual or authorities.

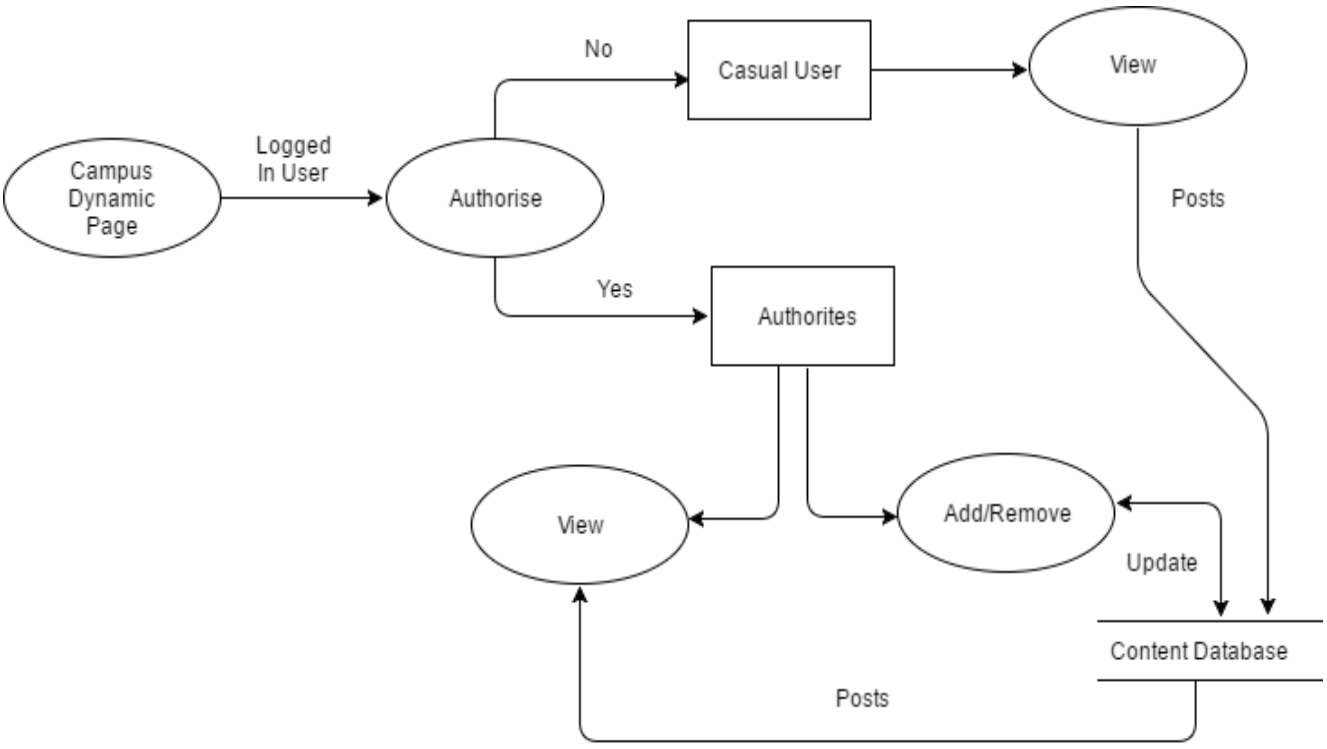


Fig 4.1.3.5

Forums:

The forums page is intended to provide a platform for problem solutions and discussion about any topic the users are interested in. Both the casual users and authorities are allowed to both view and edit the page. The data created on the forum pages are updated in the database.

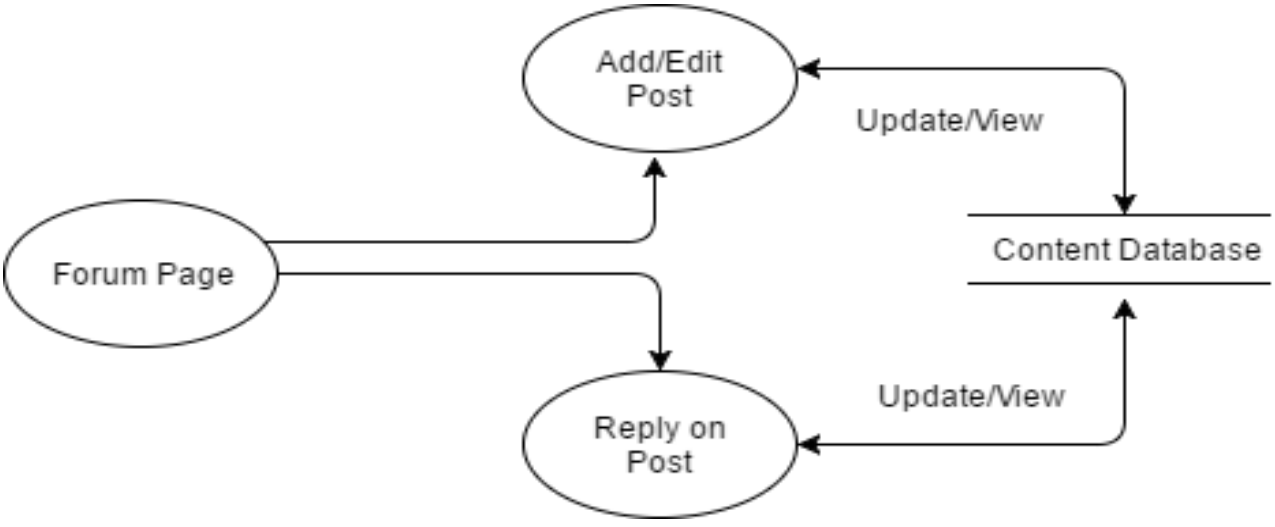


Figure 4.1.3.6

Search:

The search is only viewing the profile, another user. The user set includes both the casual users and authorities. The user can enter any details of the user he/she have to search in the search bar provided. The detail will be searched on the database and if the user is found then details of the user will be displayed on the user page.

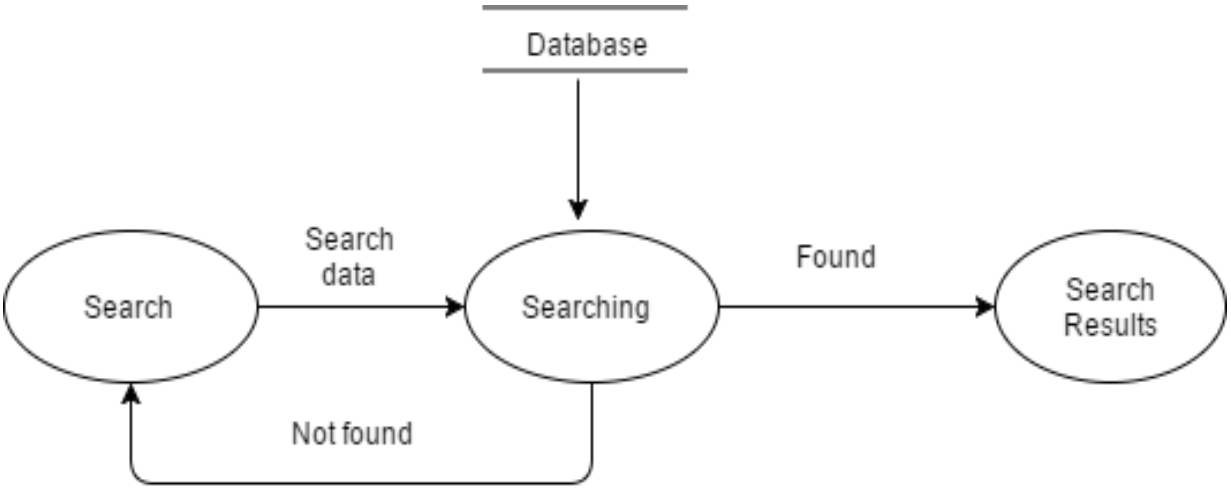


Figure 4.1.3.7

Profile:

The user profile contains the details user provided on registration. The user can both view and edit the details when viewed the data will be provided from the database and the edited details are updated to the database.

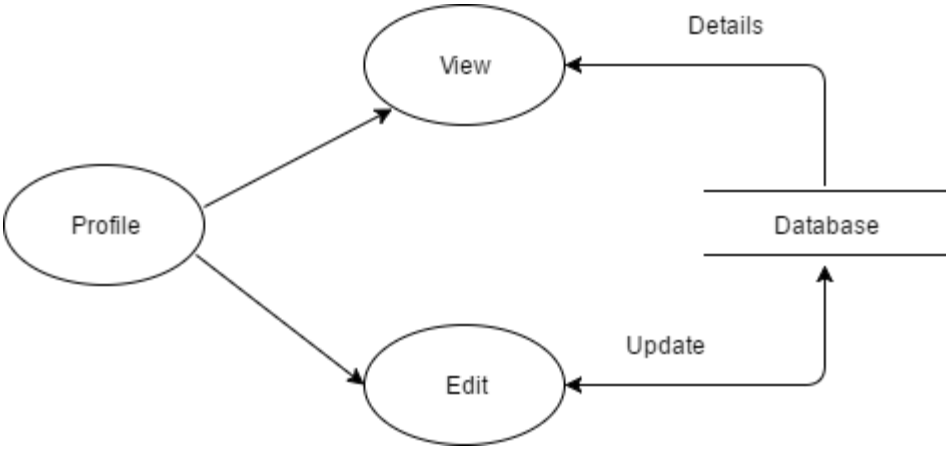


Figure 4.1.3.8

4.2 ENTITY – RELATIONSHIP DIAGRAM

The entity relationship model is a high-level data model. It is based on a perception of a real world that consists of a collection of basic objects, called entities, and of the relationshipbetween these objects. It was developed to facilitate database design by allowing specification of an enterprise schema, which represents the overall logical structure of a database.

Entity: An entity is an object that has its existence in the real world. It includes all those "things" about which data is collected. An entity may be a tangible object such as a student, a place or a part. It may also be non-tangible such as an event, a job title or a customer account. For example, if we say that a customer buys goods, it means customer and goods are entities. Diagrammatically, entities are represented by rectangles.

An Entity Set: It is a set of entities of the same type that share the same properties, or attributes. The set of all persons who are customers at a given bank, example, can be defined as the entity set customer.

Attributes: Attributes are units that describe the characteristics or properties of entities. In a database, entities are represented by tables and attributes by columns. For example, a customer entity might have numerous attributes such as code, name, and addresses. Similarly, the goods entity may have attributes like code and price. They are drawn in elliptical shapes along with the entity rectangles.

The entity-relationship diagram is shown in figure 4.2.1,

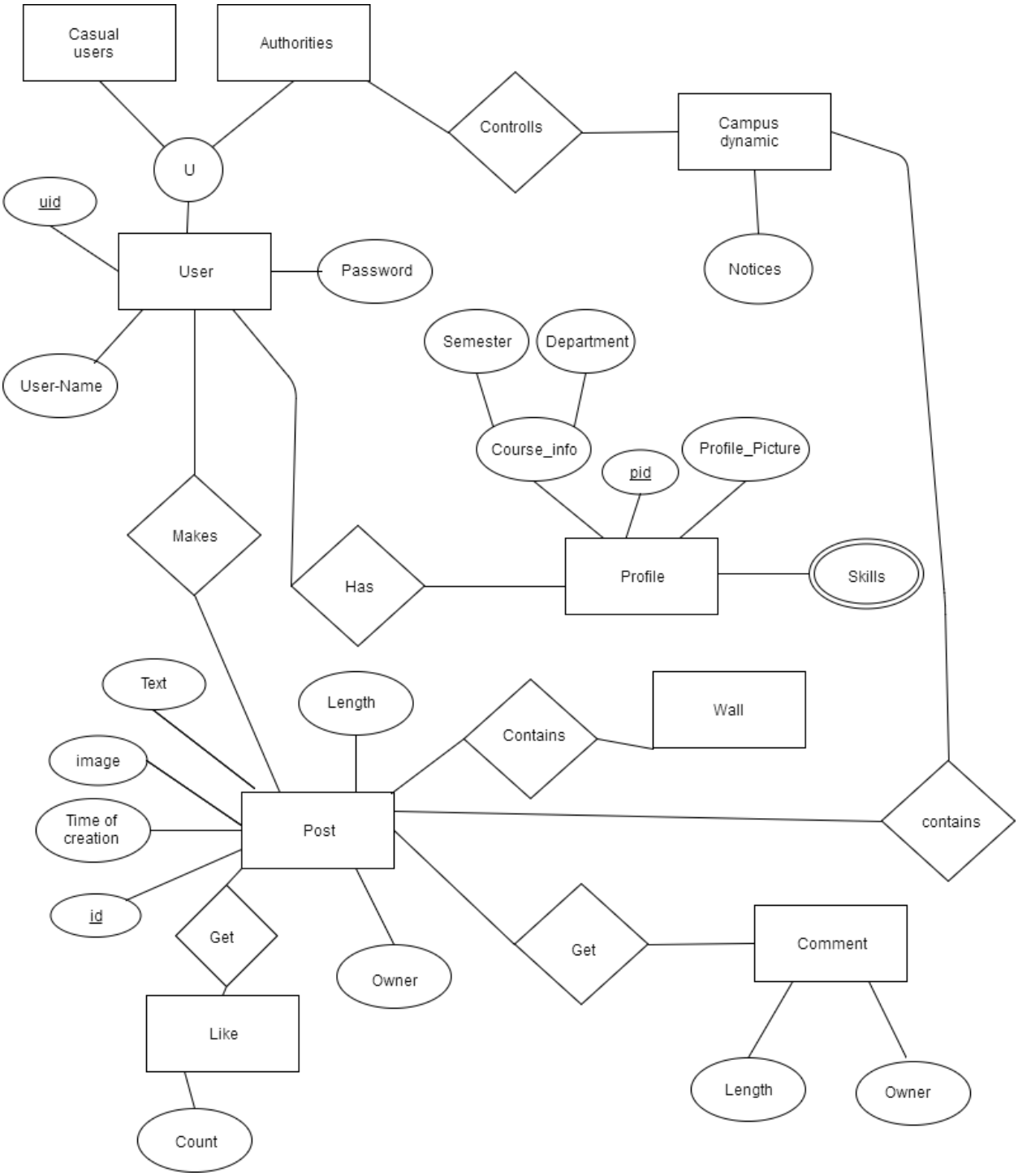

Casual users
Authorities
Controlls
Campus dynamic
U
uid
User
Password
Notices
User-Name
Semester
Department
Course_info
pid
Profile_Picture
Makes
Has
Profile
Skills
Text
Length
Contains
Wall
image
Time of creation
Post
contains
id
Get
Owner
Get
Comment
Like
Length
Owner
Count

Figure 4.2.1

4.3 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. A use case diagram contains four components.

- The actors
- The use case
- The relationships

The use case diagram is shown below in figure 4.3


Registration
Login
View wall
Edit wall
Like/Comment on wall
View dynamic
Edit dynamic
View profile
Edit profile
search profile
View forms
Edit forms
Replay on forms
casual users
Authorities

Figure 4.3

4.4 SEQUENCE DIAGRAM

A sequence diagram provides a graphical representation of theinteraction between objects overtime. It shows the message that passes between classes overt time for a use case. The objects involved in the operations are listed from left to right according to when they take part in the message sequence. A sequence diagram typically shows a user or an actor, and the objects and components they interact with in the execution of a use case.
There are four primary elements of a sequence diagram:

- Objects: are arranged on the horizontal axis. Time increases downwards. Below each object, there is a vertical dashed line. That is the life-line of the object spanning the period (the vertical axis represents time) over which the object lives.
- Lifelines: The life-line of the object spanning the period (the vertical axis represents time) over which the object lives.
- Messages: Between the life lines (or activity bars) we show arrows representing the messages which are exchanged between objects. The message itself is shown as a label on the message arrow.
- The focus of control: These are vertical rectangular boxes used to connect and send messages between two objects in a sequence diagram. It is also used to bridge the communication of different lifelines.

The sequence diagram of the Cambuzz is shown in figure 4.4

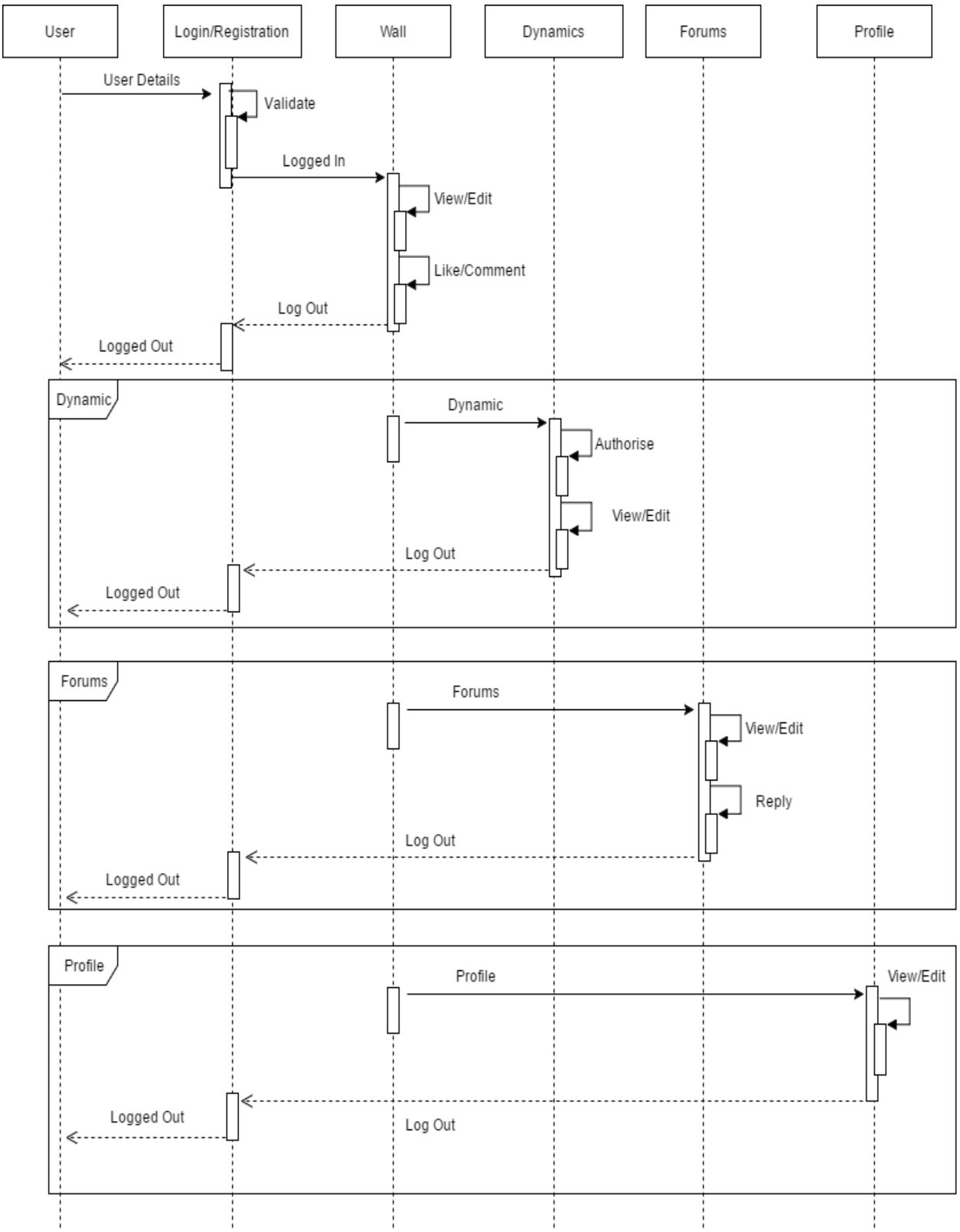

User
Login/Registration
Wall
Dynamics
Forums
Profile
User Details
Validate
Logged In
View/Edit
Like/Comment
Log Out
Logged Out
Dynamic
Dynamic
Authorise
View/Edit
Log Out
Logged Out
Forums
Forums
View/Edit
Reply
Log Out
Logged Out
Profile
Profile
View/Edit
Logged Out
Log Out

Figure 4.4

4.5 CLASS DIAGRAM

A UML Class Diagrams is a type of static structure diagram that is used both for general conceptual modeling of the systematics of the application and for detailed modeling translating the models into programming code. It describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. There are basically five 6 classes can be specified: User, Login, Wall, Campus Dynamic, forums, profile. The class diagram is shown in figure 3.5.1

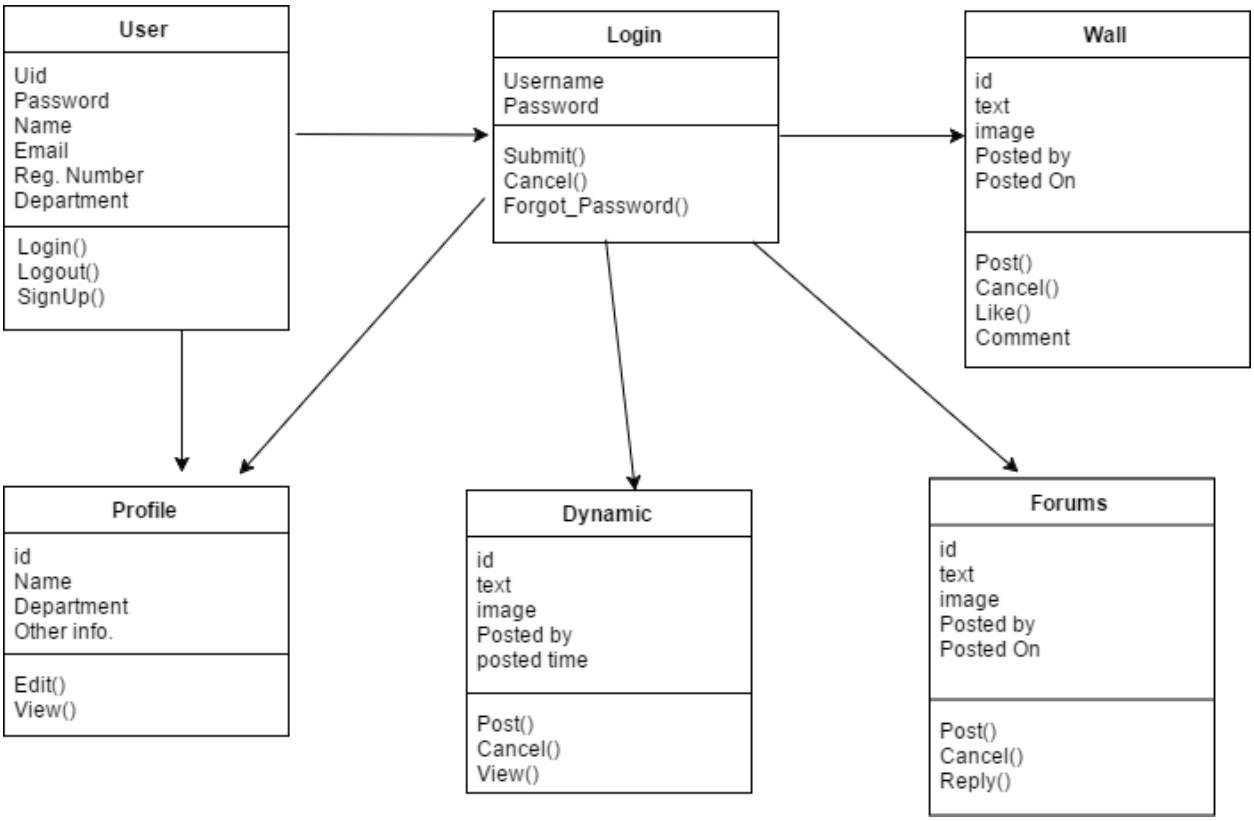


Figure 4.5

5 IMPLEMENTATION

5.1 INTRODUCTION

Implementation is the stage of the project when the theoretical design is turned into a working system. Thus it can be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.
The project implements PHP, MySQL, and standard HTML. The project will be capable of running on any standard internet web browsers.

5.2 MODULE IMPLEMENTATION

This project is an in campus social network web application for all the students and staffs in the college. The modules are divided according to the webpage they belong. The main modules are as follows

5.2.1 SIGNUP MODULE

A new user must create an account by filling the required fields of the signup form. The authority users are provided with an account so they do not require to signing up for a new account. The signup page also contains the login module; the page is developed in HTML and PHP. The entry contains basic details of the user. All the fields in the signup form are mandatory. Creation of multiple accounts is prevented by checking whether the user has entered the combination of username and email fields that are already existing in the database. The account details of the users are stored in the Users table in the database. Using a md5 encryption technique the password will be encrypted so that the password remains confidential.
Once the registration is complete the user will be redirected to a page home page.

Main functions used in this module are:
mysql_connect(<servername>,<username>,<password>);
to create a connection to the database, mysql_connect is used

mysql_select_db(<database name>,< result of mysql_connect()>);
To select a database, mysql_select_db() is used.

mysql_query("<sql statement>");

To execute the mysql queries in PHP, mysql_query() function is used

mysqli_num_rows(<variable>);

To count the number of rows

getElementById();

Search for an element with a certain id

sessio_start();

To start a user session

id	username	First_name	Last_name	email	password	Signup_date	activated	bio	Profile_pic

Table 5.1 User

5.2.2 LOGIN MODULE

Registered users provide username and password to login into their account. Both the casual and authority user uses the same login page. The login page is developed using HTML / PHP. For casual users they must enter the username and password they provided during the creation of their new account. And the authority user can use the username and password they are provided. Once the user clicks the submit button the user details are validated using the uses table in the database. Once they found valid they are successfully redirected to their home page.

Main functions used in this module are:

mysql_connect(<servername>,<username>,<password>)

To create a connection to the database, mysql_connect is used

mysql_select_db(<database name>,< result of mysql_connect()>);

To select a database, mysql_select_db() is used.

mysql_query("<sql statement>");

To execute the mysql queries in PHP, mysql_query() function is used

mysqli_num_rows(<variable>);

To retrieve the contents of rows of a particular table from the database

mysqli_fetch_array();

Fetch a result as an associative array

getElementById();

Search for an element with a certain id

sessio_start();

To start a user session

Login module uses the same users table described in the signup module.

5.2.3 HOME MODULE

After the successful login, the users are redirected to the home page. For authority users, the home page is where the posts posted on dynamic pages are uploaded. For both the users he home page provide a brief description about the cambuzz and the about the college.

The authority user can upload posts by using the text field provided and photos can also be uploaded. For text posts, the text field must be filled and by clicking the post button the dynamic table gets updated with the body of the post with other required rows of the table. The photo can be uploaded by using the browse option for selecting the picture and then clicking the post button. The photo will be stored in a randomly generated folder within the dynamic_post directory in a randomly generated name. The id or the name of the folder/ the picture will be stored in the post_pic field in the dyanamic_posttable. The posts that get uploaded from the home field by authority users will be displayed in the dynamic page of the user.

The main functions used in this module are

mysql_connect(<servername>,<username>,<password>);

to create a connection to the database, mysql_connect is used

mysql_select_db(<database name>,< result of mysql_connect()>);

To select a database, mysql_select_db() is used.

mysql_query("<sql statement>");

To execute the mysql queries in PHP, mysql_query() function is used

mysql_fetch_assoc(<stirng>);

To retrieve the row associated with the specified string

id	body	Date_added	Added_by	User_posted_to

Table 5.2 Dynamic

5.2.4 PROFILE MODULE

Both the users have the profile page, which displays the basic details they provided during the registration, edit profile button and a post module for posting on their wall page.
The details section consists of a profile picture and the name, email, and skills set. These details are retreivedfrom the table user from the database once the user is logged successfully.

The edit profile button redirects the user to theaccount_setting page. The user can change their profile picture by selecting the picture by the browse option for selecting the picture and then clicking the upload button. The photo will be stored in a randomly generated folder within the post directory in a randomly generated name. The id or the name of the folder/ the picture will be updated in the profile_pic field in the users table.

The text fields are provided to update some basic details about the user including the first name, last name, and a skill set. The update will be updated in the users table as well as on the profile page of the user. Once the user creates a new user password the password will get encrypted and will be updated in the users table. The user can return to the profile page using the back button.

The main functions used in this module are

mysql_connect(<servername>,<username>,<password>);
to create a connection to the database, mysql_connect is used

mysql_select_db(<database name>,< result of mysql_connect()>);
To select a database, mysql_select_db() is used.

mysql_query("<sql statement>");
To execute the mysql queries in PHP, mysql_query() function is used

mysql_fetch_assoc(<stirng>);
To retrieve the row associated with the specified string

mysqli_num_rows(<variable>);
To count the number of rows in a result

5.2.5 WALL MODULE

The wall module consists of the posts that are posted by all other users of the Cambuzz. The posts are uploaded from the profile page where the post module is integrated. On the profile page, the user can enter the texts using the text field provided on the profile page and when the post button is clicked the post will be uploaded to the post table in the database and it is displayed just below it. On the wall page, all the posts are displayed in the order in which the newest first. The posts body is retrieved form the posts table with the user details. The posts can be liked or commented by any valid user.

The comment module consists of a comment button which is a scrollable list. The list displays all the comments done previously. The comments are retrieved form the post_comments table in the database in order in which the newest comes on top. The user can comment on the post by

entering the text in the text field provided on the top of the comments. Once the comment button is clicked the data will be uploaded to the table post_comments on the database.

The main functions used in this module are

mysql_query("<sql statement>");

To execute the mysql queries in PHP, mysql_query() function is used

my_sql_fetch_assoc(<stirng>);

To retrieve the row content associated with a field

mysqli_num_rows(<variable>);

To count the number of rows in a result

5.2.6 FORUM MODULE

The forum module consists of posts posted by any valid user. It is page intended for open discussion about any topic. The posts can be uploaded from the forum page itself. The user can upload both the photos and texts as a post. The text field is provided with a post button. The posted text data will be stored in the forum_post table in the database. The pictures uploaded by the users will be saved with a random name in the forum directory and the name of the photo is uploaded in the forum table in the post_pic field.

The posts are provided with the option to reply to it. Any user can post his/her own reply in the reply filed provided. The posted reply will be stored in the forum_post table in the appropriate field in the database. The replies are updated with the posts and are public.

The main used in this module are

mysql_query("<sql statement>");

To execute the MySQL queries in PHP, mysql_query() function is used

my_sql_fetch_assoc(<stirng>);

To retrieve the row content associated with a field

5.2.7 DYNAMIC MODULE

The dynamic module consists of posts that are posted by authority users. Since the posts are posted form the home page, the posts are fetched from the dynamic table in the database and shown on the page itself.

The main functions used in this module are

mysql_query("<sql statement>");

To execute the mysql queries in PHP, mysql_query() function is used

my_sql_fetch_assoc(<stirng>);

To retrieve the row content associated with a field

mysqli_num_rows(<variable>);

To count the number of rows in a result

5.2.8 LOGOUT MODULE

Once the user clicks the logout button the user session will be destroyed andthe user will be redirected to the login page.

The main functions used in this module are

session_start();

Creates a session or resumes the current one.

session_destroy();

Destroys all the data associated with the current session.

6 TESTING

6.1 INTRODUCTION

Testing is the process of running a system with the intention of finding errors. The main aim of testing is to confirm that the software product works in conformance with the specified requirements and it is a critical element of quality assurance and represents the ultimate review of analysis, design, and coding. Test case design focuses on a set of techniques for the creation of test set because that met the overall design objectives. The scope of system testing should include both manual and computerized operations

A test approach tells about the test strategy implementation of a project and defines how testing would be carried out. The approach might be Proactive where test design process is initiated as early as possible in order to find and fix the defects before the build is created or Reactive where the testing is not started until after design and coding are completed. This project uses proactive approach.

6.2 TESTING METHODOLOGY

6.2.1 TEST APPROACH

This project uses a Dynamic testing approach. The Dynamic testing includes techniques - unit testing, integration testing and system testing. Under unit testing, individuals units or individual modules are tested by the developers which involve testing of source code. Unit testing focuses on the smallest unit of the software design. This is also known as module testing. The module testing is carried out during programming stage and each module is found to be working satisfactorilywith regard to the expected output of the module. In integration testing, individual modules are grouped together and tested by the developers. Here, the purpose is to determine that modules are working as expected once they are integrated. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting atest to uncover errors associated with the interface. System testing to check whether the system works accurately and efficiently before live operation commences.

Further, it uses Web testing for the web modules. Web Testing in checking your web application for potential bugs before it’s made live or before code is moved into the production environment. During

this stage issues such as that of web application security, the functioning of the site, its access to users and its ability to handle traffic is checked. In includes Functionality testing, Usability testing, Interface testing, Database testing, Compatibility testing etc.

6.2.2 UNIT TESTING

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with the functional correctness of the standalone modules. The main aim is to isolate each unit of the system to identify, analyze and fix the defects. Unit testing in the project uses mainly Functionality testing and Database testing. Functionality testing is used to check all links in the web pages are working correctly and to make sure there are no broken links especially those in the navigation bar and also to check test forms working as expected. The database is one critical component of the project.

Database testing here checks whether data integrity is maintained while creating, updating or deleting data in the database.

Test cases for unit testing includes Table 6.1, Table 6.2, Table 6.3, Table 6.4, Table 6.5, Table 6.6, Table 6.7, Table 6.8, Table 6.9, Table 6.10, Table 6.11, Table 6.12, Table 6.13, Table 6.14, Table 6.15, Table 6.16, Table 6.17.

The Test Cases for unit testing are as follows.

Test Case ID	:	C_1.1
Test Case	:	Authorities-Login
Test Title	:	Verify Authority login with valid username and password
Description	:	Test the login page
Pre-Condition	:	Authorities has valid username and password

Test Steps	Test Data	Expected Results	Actual Results
Authority user enters a valid username and password.	Username: 1000 Password: authority_1	Redirects to Cambuzz home page	Redirects to Cambuzz home page

Table 6.1 Test Case for Authority login with username and password

Test Case ID	:	C_1.2
Test Case	:	Authorities-Profile page
Test Title	:	Authority- Profile page
Description	:	Test the profile page
Pre-Condition	:	Authority user should be successfully logged in and Selected the profile page from the navigation bar

Test Step	Test Data	Expected Output	Actual Output
User selects the profile page from the navigation bar	User mouse clicks	User is redirected profile page displaying the details of user profile	User is redirected profile page displaying the details of user profile
User enters a post text and clicks post	Text = This is my first post User mouse clicks	The post with the time and date of the post is displayed below.	The post with the time and date of the post is displayed below
User deletes a post	User mouse clicks	The post is removed from the page.	The post is removed from the page.

Table 6.2 Test case for Authority profile page

Test Case ID : C_1.3
Test Case : Authorities-Profile
Test Title : Authorities-Profile update page
Description : Test the Profile update page
Pre-Condition : Authority user should be successfully logged in and selected the profile update option from profile page

Test Step	Test Data	Expected Output	Actual Output
User selects the profile picture and clicks upload	Img=./img/profile.png	The window for selecting the picture opens up and the picture appears in the box above the buttons. Upload button redirects to profile page.	The window for selecting the picture to open up and the picture appears in the box above the buttons. Upload button redirects to profile page.
User enters the profile info and clicks update	First name = College Last name = Union Your skills = NULL	A pop-up box reading the profile update successfully pops up.	A pop-up box reading the profile update successfully pops up.
User password change	Old password = 12345 New password = 54321 Confirm password = 54321	A pop-up box reading the password is successfully changed pops up.	A pop-up box reading the password is successfully changed pops up.
User go back to the profile page	User mouse clicks	Clicks the back button and redirected to the profile page.	Clicks the back button and redirected to the profile page.

Table 6.3 Test case for Authority account settings page

Test Case ID : C_1.4
Test Case : Authorities - Home
Test Title : Authorities - Home page
Description : Test the home page
Pre-Condition : Authority user should be successfully logged in and selected the profile update option from profile page

Test Step	Test Data	Expected Output	Actual Output
User is redirected after successful login	Null	The user will be redirected to a page including the details of the cambuzz.	The user will be redirected to a page including the details of the cambuzz.
User posts on dynamic page	Text = This is my first post User mouse clicks	The post with the time and date of the post is updated in the database and displayed in the dynamic page.	The post with the time and date of the post is updated in the database and displayed in the dynamic page.

Table 6.4 Test case for home page

Test Case ID : C_1.5
Test Case : Authorities - wall
Test Title : Authorities – wall page
Description : Test the wall page
Pre-Condition : Authority user should be successfully logged in and selected the wall page from navigation bar

Test step	Test data	Expected Outputs	Actual Outputs
User selects the profile page from the navigation bar	User mouse click	The user is redirected wall page displaying the posts posted by users.	The user is redirected wall page displaying the posts posted by users.
User likes a post	User mouse click	The database is updated and the count of the increments by one.	The database is updated and the count of the increments by one.
User comments on the post	Text = this is a comment User mouse click	The comment is updated in the database and the comment is displayed with the date and time.	The comment is updated in the database and the comment is displayed with the date and time of the comment.

Table 6.5 Test case for wall page

Test Case ID : C_1.6
Test Case : Authorities - Dynamic
Test Title : Authorities –Dynamic page
Description : Test the Dynamic page
Pre-Condition : Authority user should be successfully logged in and selected the Dynamic page from navigation bar

Test Step	Test Data	Expected Outputs	Actual outputs
User selects the Dynamic page from the navigation bar	User mouse click	The user is redirected Dynamic page displaying the posts posted by Authority users.	The user is redirected Dynamic page displaying the posts posted by Authority users.

Table 6.6 Test case for Dynamic page

Test Case ID : C_1.7
Test Case : Authorities - Forum
Test Title : Authorities – Forum page
Description : Test the Forum page
Pre-Condition : Authority user should be successfully logged in and selected the Forum page from navigation bar

Test Case	Test Data	Expected Outputs	Actual Results
User selects the Forum page from the navigation bar	User mouse click	The user is redirected Forum page displaying the posts posted by Authority users.	The user is redirected Forum page displaying the posts posted by Authority users.
Post on forum	Text = this is a test? User mouse click	The post is updated on the database and displayed on the field below.	The post is updated on the database and displayed on the field below.
Reply to posts	User mouse clicks Text = this is a reply	The reply text is updated in the database and shown in the reply list.	The reply text is updated in the database and shown in the reply list.

Delete the post	User mouse clicks	The post deleted is removed from the database and also removed from the forum page.	The post deleted is removed from the database and also removed from the forum page.

Table 6.7 Test case for Forum page

Test Case ID : C_1.8
Test Case : Authorities - Logout
Test Title : Authorities – Logout
Description : Test the Logout
Pre-Condition : Authority user should be successfully logged in and selected the Logout from navigation bar

Test Case	Test Data	Expected Outputs	Actual Outputs
The user selects the logout button for logout the session.	User Mouse clicks	The session is ended and the user is redirected to login page.	The session is ended and the user is redirected to login page.

Table 6.8 Test case for Logout

Test Case ID : C_1.9
Test Case : Casual User - Signup
Test Title : Casual User – Signup for
Description : Test the Signup form

Test Case	Test Data	Expected Outputs	Actual Outputs
User enters all the required details in signup form and clicks signup button	Details corresponding to all the fields are entered	Redirects to page containing link to the login page.	Redirects to page containing link to the login page.

Table 6.9 Test case for Signup

Test Case ID : C_1.10
Test Case : Casual User - Login
Test Title : Verify Casual user login with valid username and password
Description : Test the login page
Pre-Condition : Casual user has a valid username and password

Test Case	Test Data	Expected Outputs	Actual Outputs
The casual user enters a valid username and password.	Username : user_1 Password: user1@	Redirects to Cambuzz home page	Redirects to Cambuzz home page

Table 6.10 Test case for Casual user login

Test Case ID : C_1.11
Test Case : Casual User - Home
Test Title : Casual User - Home page
Description : Test the home page
Pre-Condition : Casual user should be successfully logged in and selected the profile update option from profile page

Test Case	Test Data	Expected Outputs	Actual Outputs
User is redirected after successful login	Null	The user will be redirected to a page including the details of the cambuzz.	The user will be redirected to a page including the details of the cambuzz.

Table 6.11 Test case for Casual user Home page

Test Case ID : C_1.12
Test Case : Casual User - Profile page
Test Title : Casual User - Profile page
Description : Test the profile page
Pre-Condition : Casual user should be successfully logged in and Selected the profile page from the navigation bar

Test Case	Test Data	Expected Outputs	Actual Outputs
User selects the profile page from the navigation bar	User mouse clicks	User is redirected profile page displaying the details of user profile	User is redirected profile page displaying the details of user profile
User enters a post text and clicks post	Text = This is my first post User mouse clicks	The post with the time and date of the post is displayed below.	The post with the time and date of the post is displayed below
User deletes a post	User mouse clicks	The post is removed from the page and database.	The post is removed from the page and database.

Table 6.12 Test case for Casual user Profile page

Test Case ID	:	C_1.13
Test Case	:	Casual User -Profile
Test Title	:	Casual User -Profile update page
Description	:	Test the Profile update page
Pre-Condition	:	Casual user should be successfully logged in and selected the profile update option from profile page

Test Case	Test Data	Expected Outputs	Actual Outputs
User selects the profile picture and clicks upload	Image=./img/profile1.png	The window for selecting the picture opens up and the picture appears in the box above the buttons. Upload button redirects to profile page.	The window for selecting the picture to open up and the picture appears in the box above the buttons. Upload button redirects to profile page.
User enters the profile info and clicks update	Details corresponding to all the fields are entered	A pop-up box reading the profile update successfully pops up.	A pop-up box reading the profile update successfully pops up.
User password change	Details corresponding to all the fields are entered	A pop-up box reading the password is successfully changed pops up.	A pop-up box reading the password is successfully changed pops up.
User go back to the profile page	User mouse clicks	Clicks the back button and redirected to the profile page.	Clicks the back button and redirected to the profile page.

Table 6.13 Test case for Casual user Account settings page

Test Case ID : C_1.14
Test Case : Casual User - Dynamic
Test Title : Casual User – Dynamic page
Description : Test the Dynamic page
Pre-Condition : Casual user should be successfully logged in and selected the Dynamic page from navigation bar

Test Case	Test Data	Expected Outputs	Actual Outputs
User selects the Dynamic page from the navigation bar	User mouse click	User is redirected Dynamic page displaying the posts posted by Authority users.	User is redirected Dynamic page displaying the posts posted by Authority users.

Table 6.14 Test case for Casual user Dynamic page

Test Case ID : C_1.15
Test Case : Authorities - wall
Test Title : Authorities – wall page
Description : Test the wall page
Pre-Condition : Authority user should be successfully logged in and selected the wall page from navigation bar

Test Case	Test Data	Expected Outputs	Actual Outputs
User selects the profile page from the navigation bar	User mouse click	User is redirected wall page displaying the posts posted by users.	User is redirected wall page displaying the posts posted by users.
User likes a post	User mouse click	The database is updated and the count of the increments by one.	The database is updated and the count of the increments by one.

User comments on the post	Text = this is a comment User mouse click	The comment is updated in the database and the comment is displayed with the date and time of the comment.	The comment is updated in the database and the comment is displayed with the date and time of the comment.

Table 6.15 Test case for Casual user Wall page

Test Case ID : C_1.16
Test Case : Casual User - Forum
Test Title : Casual User– Forum page
Description : Test the Forum page
Pre-Condition : Casual user should be successfully logged in and selected the Forum page from navigation bar

Test Case	Test Data	Expected Outputs	Actual Outputs
User selects the Forum page from the navigation bar	User mouse click	The user is redirected Forum page displaying the posts posted by Authority users.	The user is redirected Forum page displaying the posts posted by Authority users.
Post on forum	Text = this is a test? User mouse click	The post is updated on the database and displayed on the field below.	The post is updated on the database and displayed on the field below.
Reply to posts	User mouse clicks Text = this is a reply	The reply text is updated in the database and shown in the reply list.	The reply text is updated in the database and shown in the reply list.

Delete the post	User mouse clicks	The post deleted is removed from the database and also removed from the forum page.	The post deleted is removed from the database and also removed from the forum page.

Table 6.16 Test case for Casual user Forum page

Test Case ID : C_1.17
Test Case : Casual User - Logout
Test Title : Casual User – Logout
Description : Test the Logout
Pre-Condition : Authority user should be successfully logged in and selected the Logout from navigation bar

Test Case	Test Data	Expected Outputs	Actual Outputs
The user selects the logout button for logout the session.	User Mouse clicks	The session is ended and the user is redirected to login page.	The session is ended and the user is redirected to login page.

Table 6.17 Test case for Casual user Logout

6.2.3 INTEGRATION TESTING

When all the development of all the units or modules is completed and integrated the integrity test phase is started. Data can be lost across an interface. One module can have an adverse effect on another, sub-functions, when combined, may not be linked in desired manner in major functions. The objective is to take unit tested modules and builds program structure. All the modules are combined and tested as a whole and to verify whether the inter-modules exchange of information and events are as per required system behavior.

Integration testing in the project uses mainly Functionality testing, Usability testing, Interface testing, Database testing, Compatibility testing. Functionality testing here includes checking the forms for whether the default values are being populated and once submitted, the data in the form is submitted to a live database. Usability testing is a vital part of any web based project. It here includes testing site navigation and content for usability. Database testing in integration testing checks whether the data retrieved from the database is shown accurately. Compatibility testing ensures that website functions correctly across browsers.

Test Case ID	:	C_2.1
Test Case	:	Signup Form
Test Title	:	Signing up
Description	:	Creating a new Casual user account
Pre-Condition	:	User should know all his/her basic details

Test Case	Test Data	Expected Outputs	Actual Outputs
User enters all the required fields in the signup form and clicks the signup button	Details corresponding to the required details are filled	Stores the details of the user in the corresponding database table, then the page is reloaded.	Stores the details of the user in the corresponding database table, then the page is reloaded.

Table 6.18 Test case for a Casual user signup

Test Case ID : C_2.2

Test Case : Login

Test Title : Authenticate Login

Description : Test Login

Pre-Condition : User should know his/her user credentials

Test Case	Test Data	Expected Outputs	Actual Outputs
User enters the user name and password and clicks the login button	Username = user_123 Password = 12345	Verifies the data entered with the data stored in the database and redirects to home page.	The information is verified using the data stored in the database. The user is redirected to the home page.

Table 6.19 Test case for a Casual/Authority user login

Test Case ID : C_2.3

Test Case : Edit - Profile

Test Title : Update profile

Description : Make changes to existing profile

Pre-Condition : User should be logged in

Test Case	Test Data	Expected Outputs	Actual outputs
User updates the user profile details and clicks the update button. And clicks the back button.	Details corresponding the fields are entered.	Updates the details in the corresponding database and redirects to profile page on clicking back button	The information is updated in the corresponding databases and the redirected to profile page on clicking the back button.

Table 6.20 Test case for a Casual/Authority user profile page

Test Case ID	:	C_2.4
Test Case	:	Upload posts
Test Title	:	Posts
Description	:	Posting posts on pages
Pre-Condition	:	User should be logged in and the image must be less than 2 mb

Test Cases	Test Data	Expected Results	Actual Results
User enters the test or upload image for posting on wall/ dynamic/ forums	The user inputs any text or selects any image by browsing the directories. (for casual users, post on dynamic page is unavailable)	The entered text is stored in the corresponding database tables or the image is stored in a folder and the id is stored in the corresponding database table. The posts are displayed on the corresponding pages	The entered text and the image are successfully stored in the corresponding databases, the posts are displayed on the corresponding pages.

Table 6.21Test case for a Casual/Authority user post upload

Test Case ID	:	C_2.5
Test Case	:	View Posts
Test Title	:	Posts
Description	:	Viewing posts posted by users on different pages
Pre-Condition	:	User should be logged in and selects the pages form the navigation bar.

Test Cases	Test data	Expected Outputs	Actual Outputs
User clicks the pages wall/ dynamic/ forum from the navigation bar	User clicks	Retrieve the information stored in the corresponding database tables and displays the posts.	The information corresponding different posts are retrieved form the database tables and displayed on the pages.

Table 6.22Test case for a Casual/Authority user viewing posts

Test Case ID : C_2.6

Test Case : Liking

Test Title : Posts

Description : Likes on posts

Pre-Condition : User should be logged in and selects the pages form the navigation bar.

Test Cases	Test Data	Expected Outputs	Actual Outputs
Use clicks on the like button	User mouse clicks	The counts of the likes are increments by one and the corresponding database table is updated.	The count is incremented by one and the corresponding databases also updated.

Table 6.23Test case for a Casual/Authority user liking posts

Test Case ID : C_2.7

Test Case : Commenting

Test Title : Posts

Description : Comments on posts

Pre-Condition : User should be logged in and selects the pages form the navigation bar.

Test Cases	Test Data	Expected Outputs	Actual Outputs
User clicks the comment button	User mouse clicks	The comments corresponding to the post is retrieved from the database tables and displayed as a scrollable list.	A scrollable list is displayed by retrieving the data from the corresponding tables.
The user inputs the comment text and clicks the comment button.	The user inputs any text and user mouse click	The text is updated on the corresponding database tables and displayed on the scrollable list	The comment is displayed on the list after the data is updated in the corresponding database tables

Table 6.24 Test case for a Casual/Authority user commenting on posts

Test Case ID	:	C_2.7
Test Case	:	Forgot Password
Test Title	:	User – forgot password
Description	:	To get a new password
Pre-Condition	:	User should know his/ her user credentials

Test Cases	Test Data	Expected Outputs	Actual Outputs
The user clicks on the forget password button on the login page and enters the required details.	Details corresponding to the fields are entered	The user details are validated using the corresponding table in the database and a randomly generated password is provided for the user	A new password is displayed on the page after the user credentials get validated using the corresponding table in the database

Table 6.24 Test case for Forgot password module

7 CONCLUSIONS

Teaching and learning practices have evolved well in last 10 years. Spurred in part by the changes encouraged by SNSs, learning activities that happen outside the classroom are acknowledged by many to be important as what happens inside. We strongly believe that our system can bring drastic changes and enhance the communication between students and staffs in the college through the use of a social networking site. We have developed the cambuzz, by satisfying all the requirements specified and all the modules of the system are tested successfully. The system offers one of a kind platform for enhancing the communication in campus

The system is found working efficiently and effectively and delivers the expected outcomes. New features can be added when the need arises with slight modification on the web app, which is relatively easy.

APPENDIX A: USER MANUAL

In order for any user to access the cambuzz, first, the cambuzz index page must be loaded onto a browser with standard internet connection.

For any new user, he /she must create an account. The index page consists of both the signup and login page. For creating an account, the details on the signup form must be filled correctly and clicking the signup button will create an account for the user and he/she will be redirected to the page. All fields in the signup form are mandatory (Screenshot B.1). The authority user doesn't require to signup because they are already provided with a valid account.

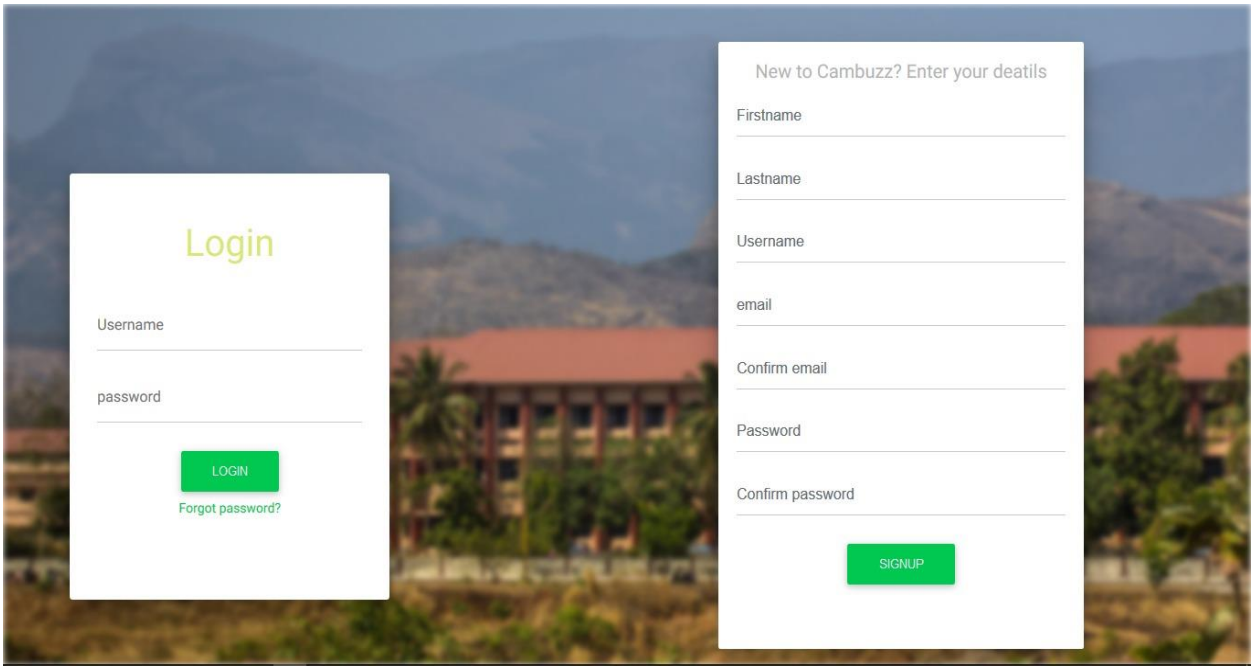
After the account has been created successfully the user is redirected to the profile page where the user can enter his/her login credentials (Screenshot B.2).

The user will be redirected to the home page after the successful login.
For the casual user, the home page contains the details of the college and Cambuzz(Screenshot B.3)
For the authority users, the posts to the campus dynamic page (Screenshot B.4). The post section consists of picture upload section and a text upload section. The user can either post a photo or can post text but not both. Once the user clicks the post button the post will be updated on the dynamic page (Screenshot B.5).

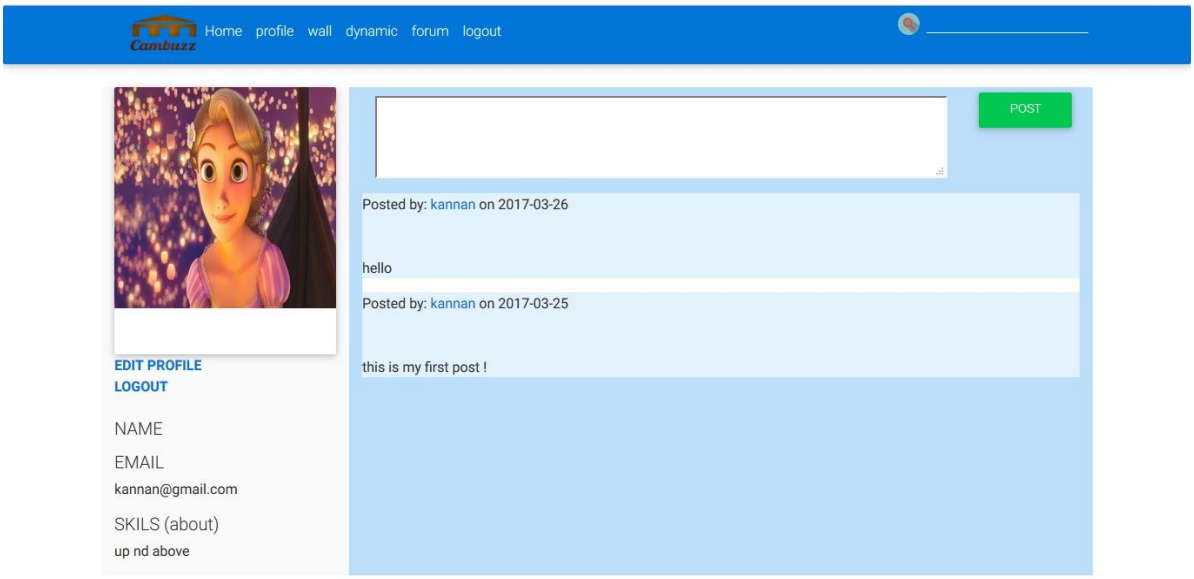
The user can navigate to any page using the navigation bar.
The profile page contains the basic information provided during creation of the account, edit profile option and a post section (Screenshot B.6).
For the user update his/her profile, the user has to click on the edit profile and will be redirected to the account settings page .
For the user to make posts to wall page, the post section in the profile page have to fill with appropriate details – picture or text and must click the post button. The post will be updated on the profile page and as well as in the wall page. (Screenshot B.7).

The wall page can be navigated using the navigation bar. The wall will be updated with the posts by every user of the Cambuzz. Any user can like or comment on the post. Once the user clicks a like button, the count of the likes will be incremented by one (Screenshot B.8).
The comments once clicked on the button will be shown as a scrollable list with a text input section at the top. The comments made by the user will be updated in real time in the list.

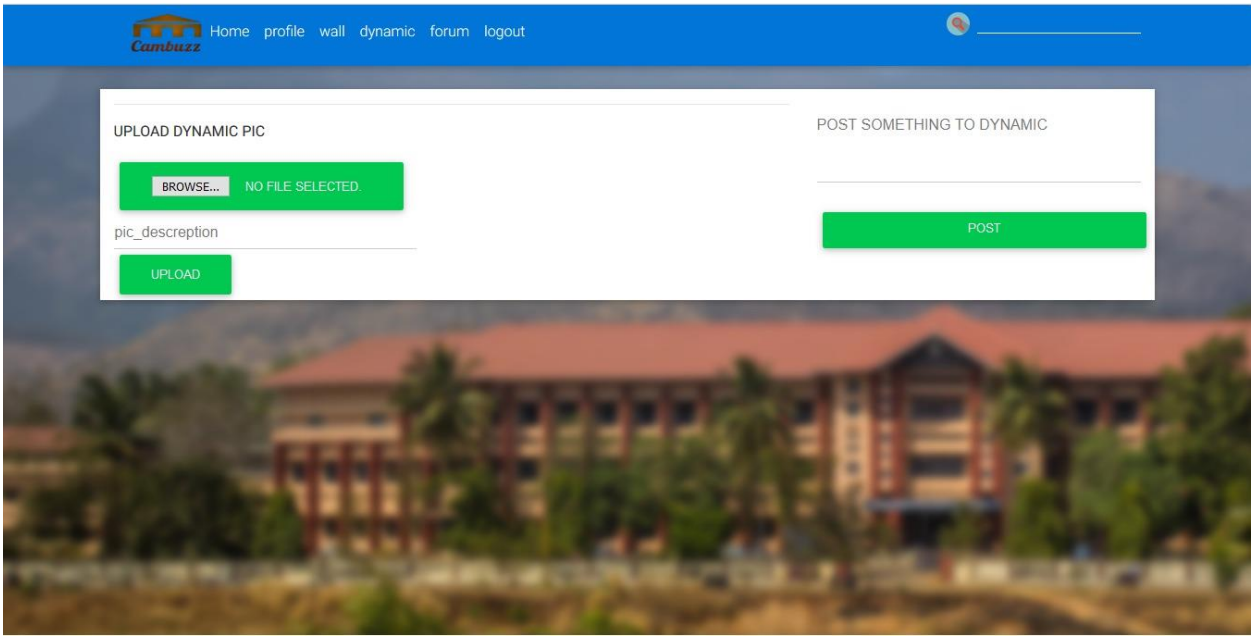
The forums page can be navigated through the navigation bar, the open discussion page with which the post section at the top of the page and the posts posted by other users shown at the bottom .The user can reply to the posts using the reply button. Just works like the comment option on the wall page (Screenshot B.9).



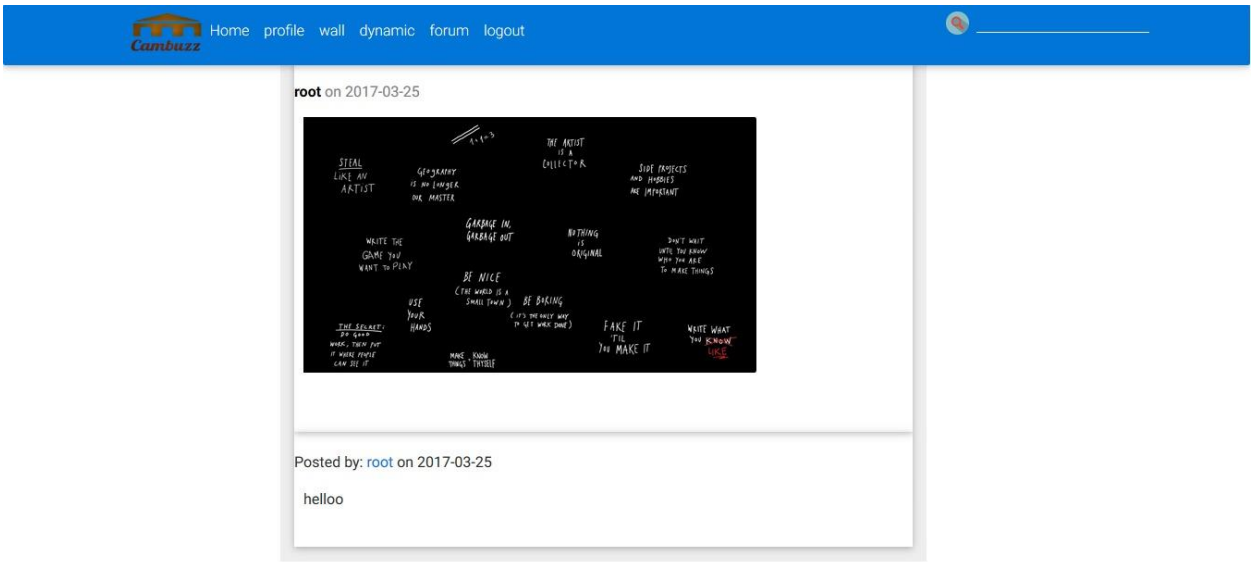
Screenshot B.1 Login/Signup page



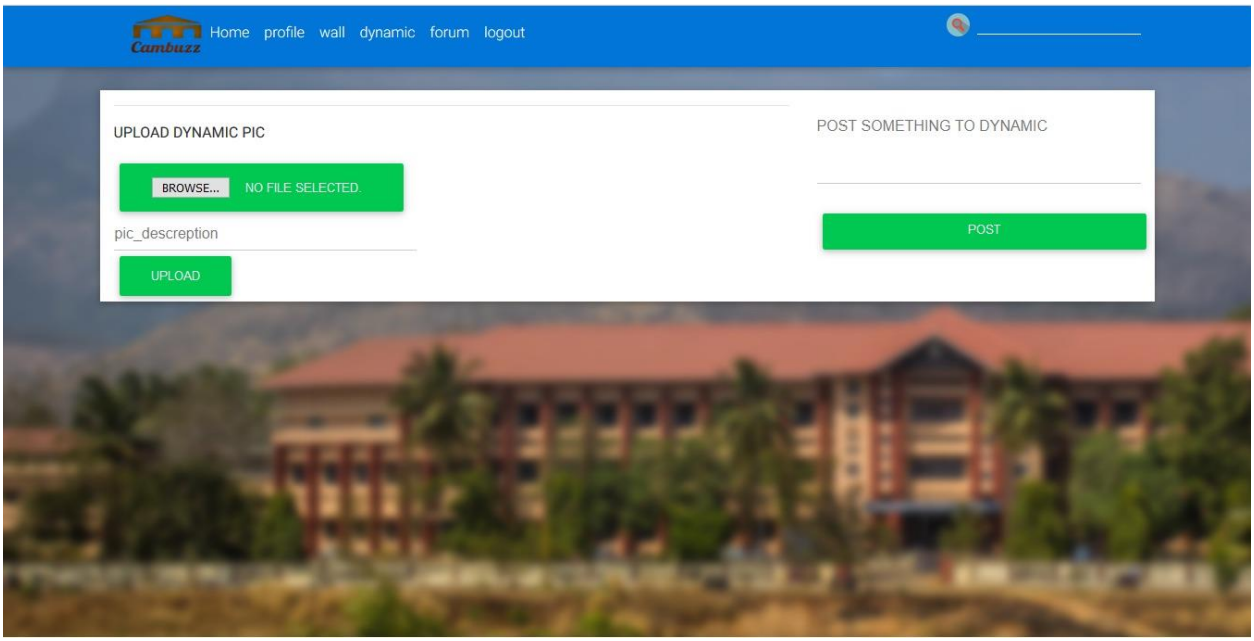
Screenshot B.2 Profile Page



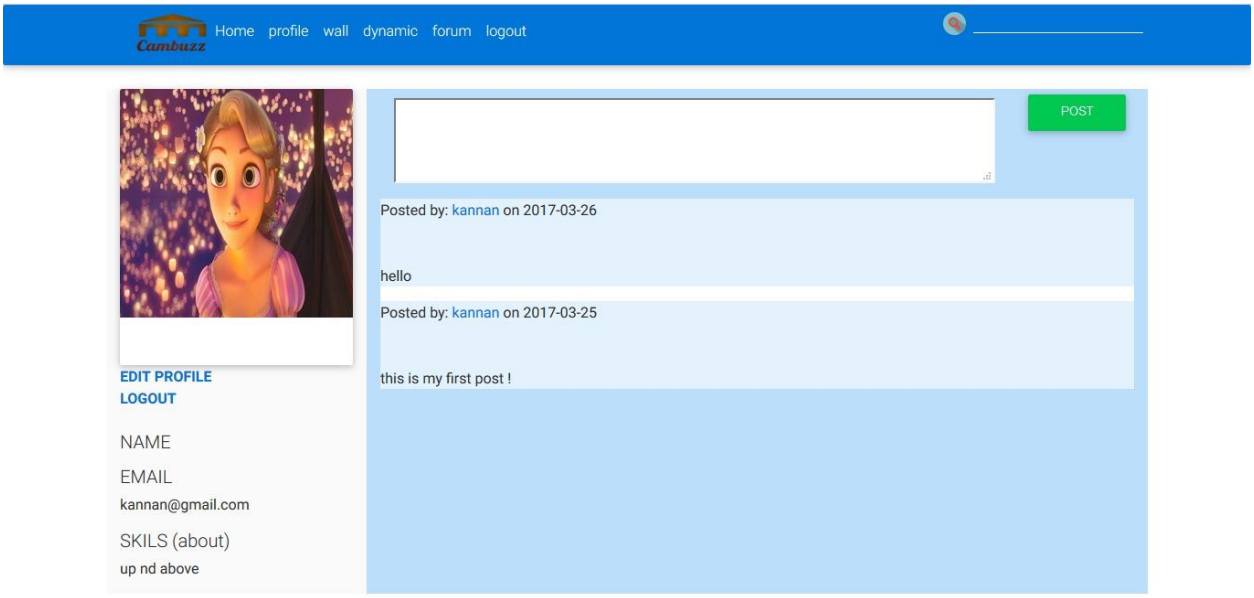
Screenshot B.3 Home page



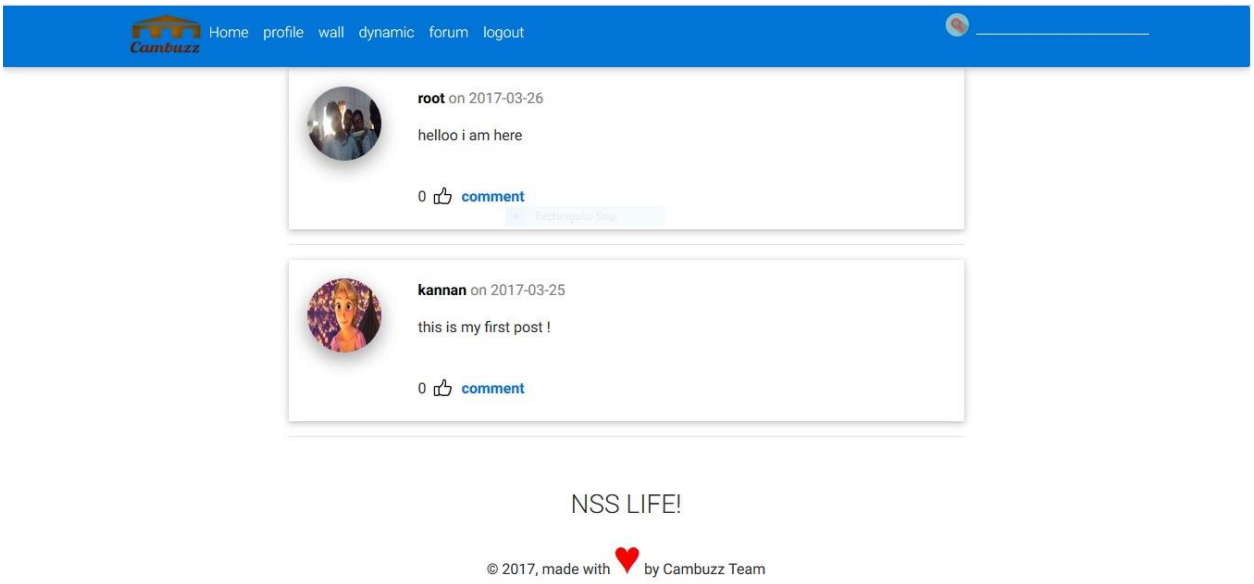
Screenshot B.4 Dynamic Page



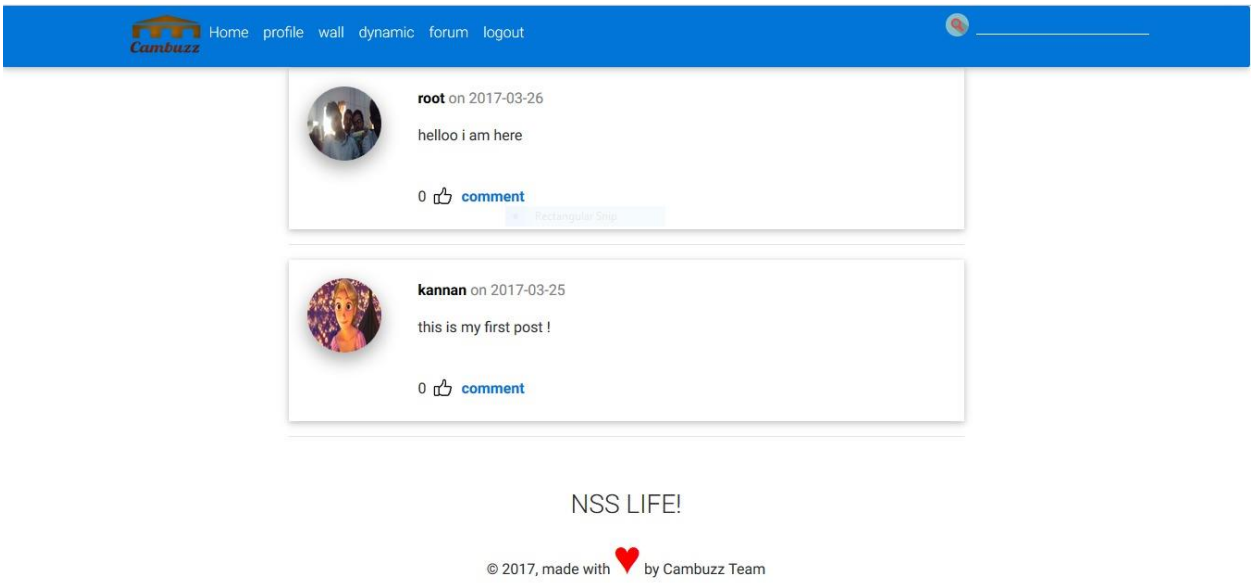
Screenshot B.5 Dynamic Page



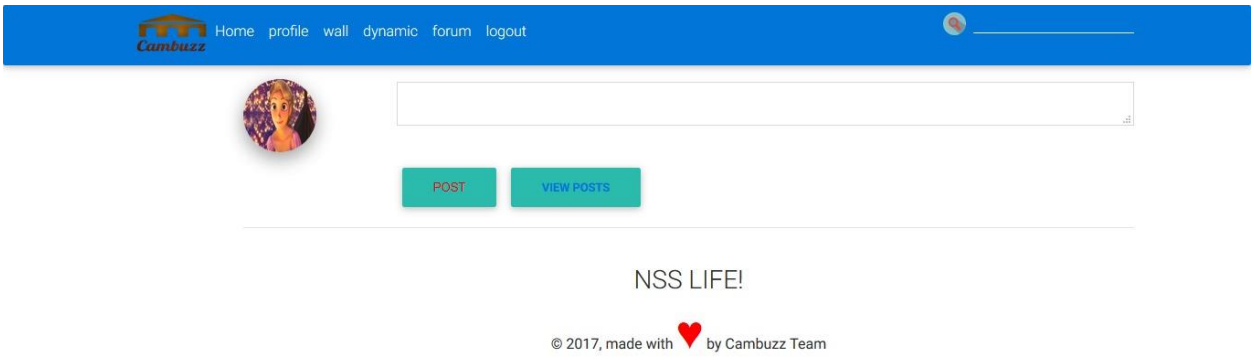
Screenshot B.6 Profile page- post Section



Screenshot B.7 Wall Page



Screenshot B.8 Wall page – Like Section



Screenshot B.9 Forum Page

REFERENCES

[1**]** **YAN Yan, XU Longzhe, YANG Renyuan** and **Wang Wanjun "***Design and Implementation of Campus Social Networking System***"**an IEEE Paper.

[2**]** **"http://www.draw.io"** *Online tool for drawing various diagrams*

[3**]** **"youtube.com"**

[4**]** **"http://www.w3schools.com"** *Online web tutorials*