



## **Detailed Design Document**

**Course Name:** Software Engineering

**Course code:** CS301

**Year:** 3<sup>rd</sup>

**Semester:** 5<sup>th</sup>

**Session:** 2018-19

**Discipline:** CSE

**Project:** NU NEWSLETTER

**Group Member:**

Aman Garg

Ajinkya Bedekar

Biren Sharma

Shantanu Bahuguna

Yogesh Sharma

# Software Design Specification

## 1. Introduction

The Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project including use case models, sequence diagrams, collaboration models, object behaviour models, and other supporting requirement information.

### 1.1 Purpose of this document

This document will define the design of the one runway simulator. It contains specific information about the expected input, output, classes, and functions. The interaction between the classes to meet the desired requirements are outlined in detailed figures at the end of the document.

### 1.2 Scope of the development project

We describe what features are in the scope of the software and what are not in the scope of the software to be developed.

#### In Scope:

- a. Website for the news taking place in NIIT University.
- b. The important events taking place in NIIT University will be published on the website.
- c. Placement news of students in NIIT University will be available in the website.

#### Out of Scope:

- a. The news/events taking place in India.

## 1.3 Definitions, acronyms, and abbreviations

IEEE: Institute of Electrical and Electronics Engineers

## 1.4 References

**1.4.1** R. S. Pressman, Software Engineering: A Practitioner's Approach, 5th Ed, McGraw-Hill, 2001.

**1.4.2** IEEE SDS template

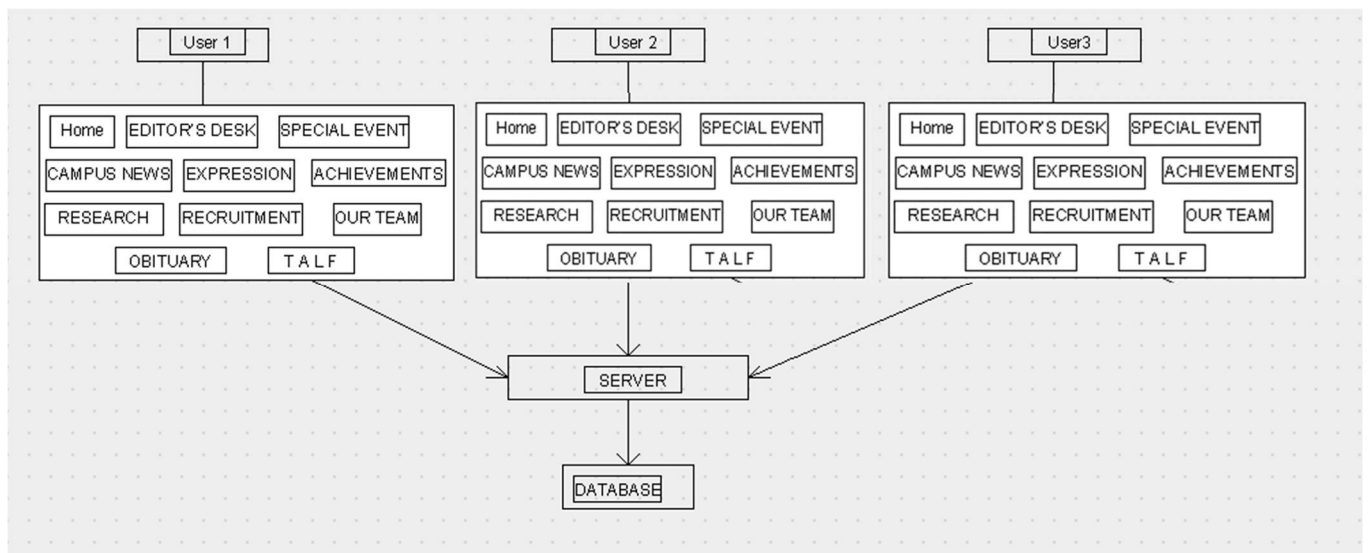
## 1.5 Overview of document

This SDS is divided into seven sections with various sub-sections. The sections of the Software Design Document are:

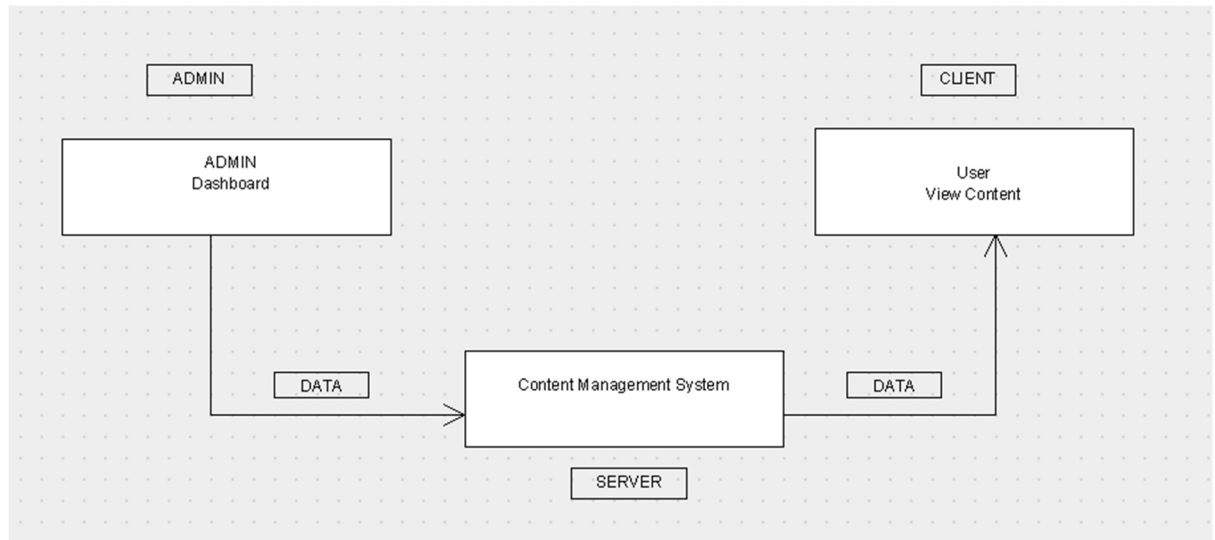
1. **Introduction:** describes about the document, purpose, scope of development, project definitions, and abbreviations used in the document.
2. **Conceptual Architecture/Architecture Diagram:** describes the overview of components, modules, structure and relationships and user interface issues.
3. **Logical Architecture:** describes Logical Architecture Description and Components.
4. **Execution Architecture:** defines the runtime environment, processes, deployment view.
5. **Design Decisions and Trade-offs:** describes the decisions taken along with the reason as to why they were chosen over other alternatives.
6. **Pseudocode for components:** describes pseudocode, as the name indicates.
7. **Appendices:** describes subsidiary matter if any.

## 2. Conceptual Architecture/Architecture Diagram

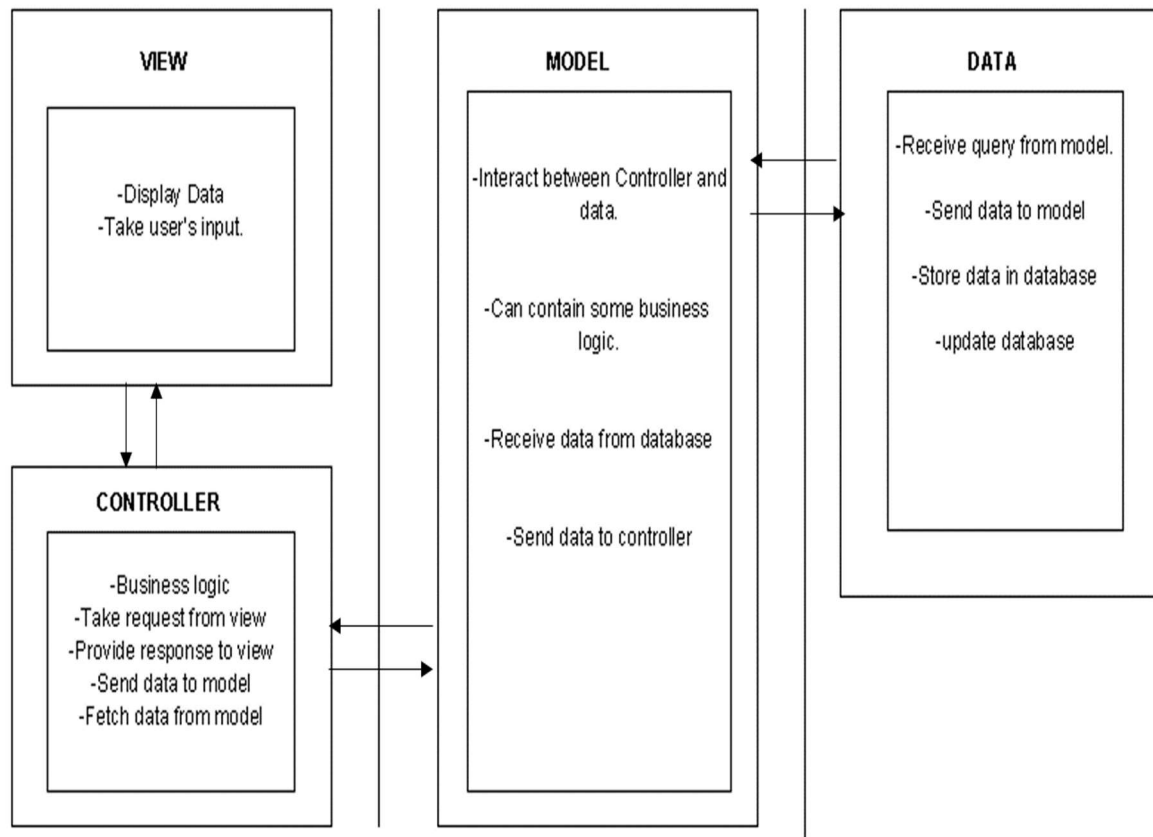
Architecture Diagram 1:



Architecture Diagram 2:

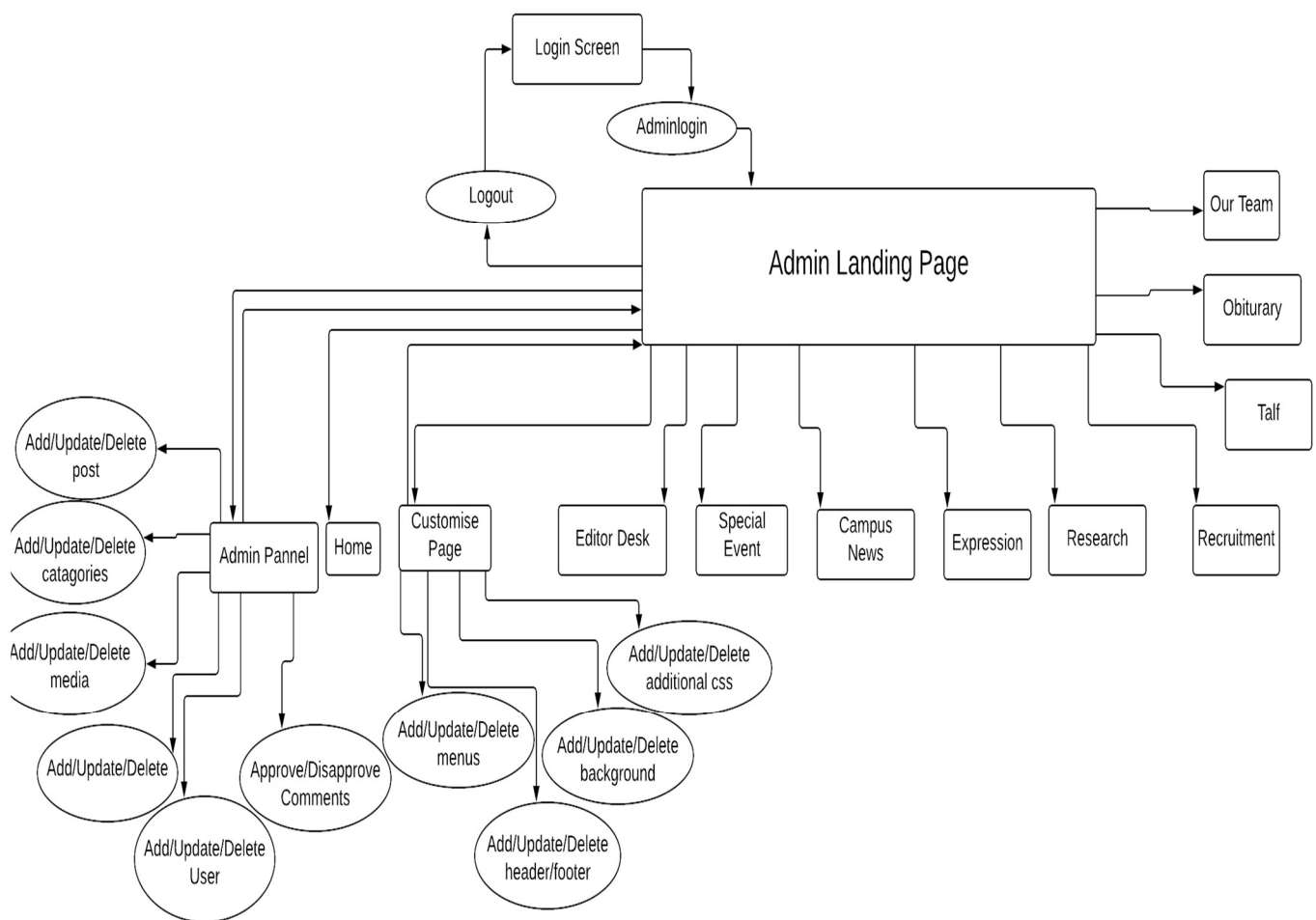


## 2.1 Overview of components



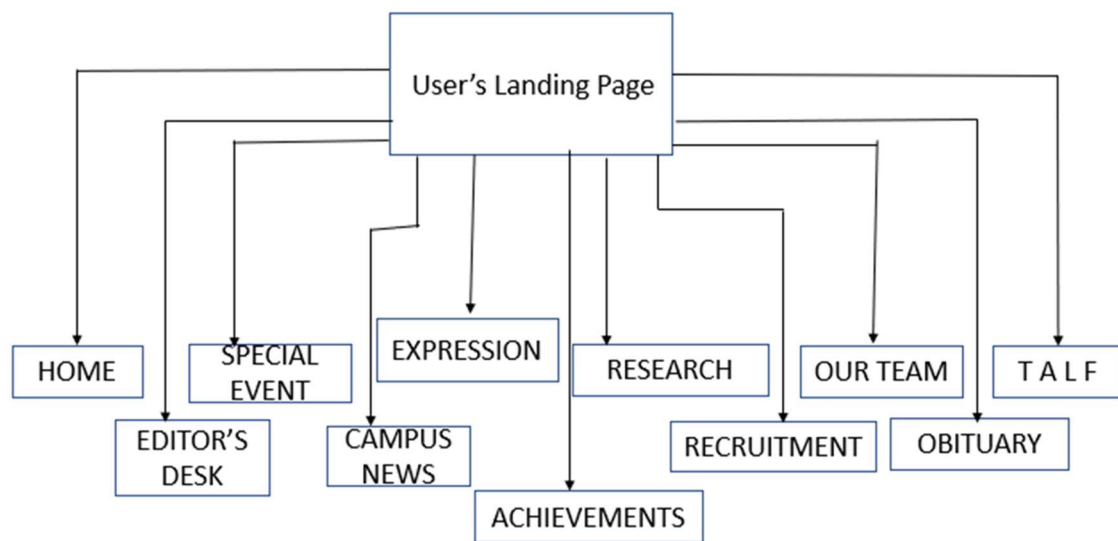
## 2.2 Structure and relationships

### 2.2.1 Admin's Side



Note: User can shift between varies tab's directly.

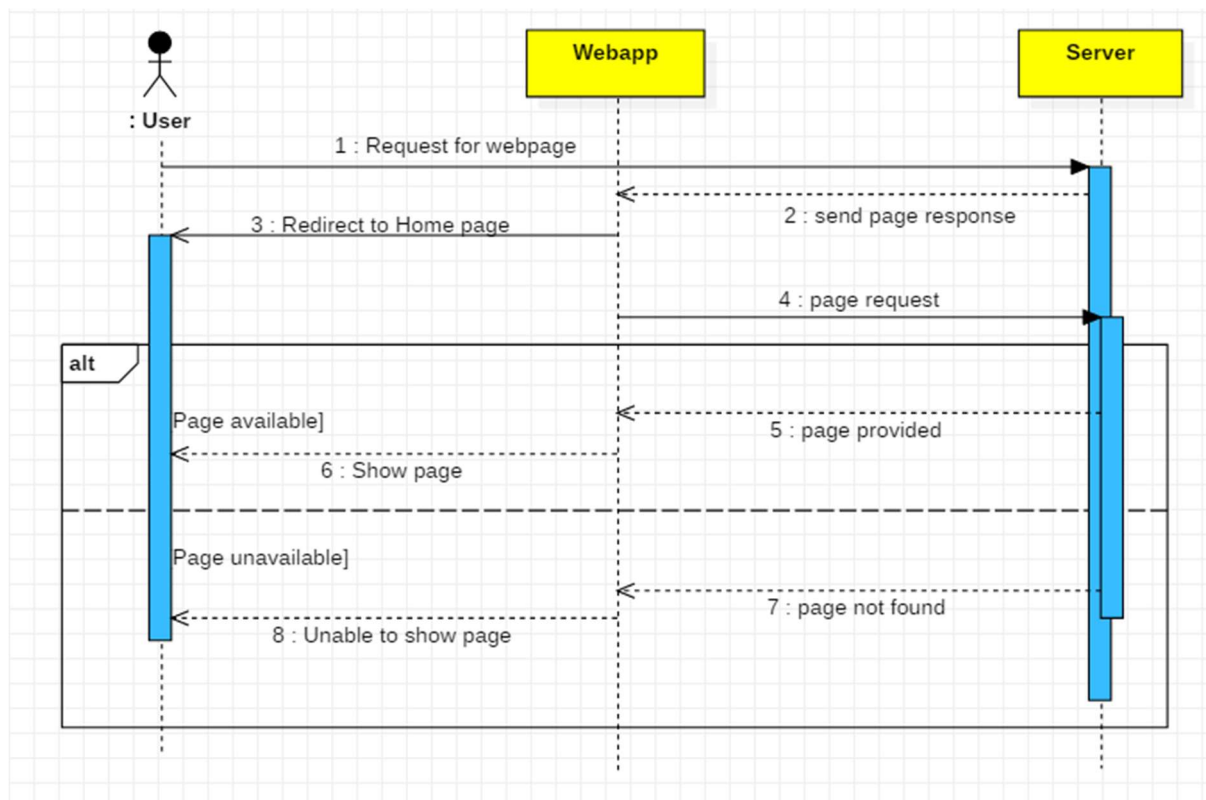
### 2.2.2 User's Side



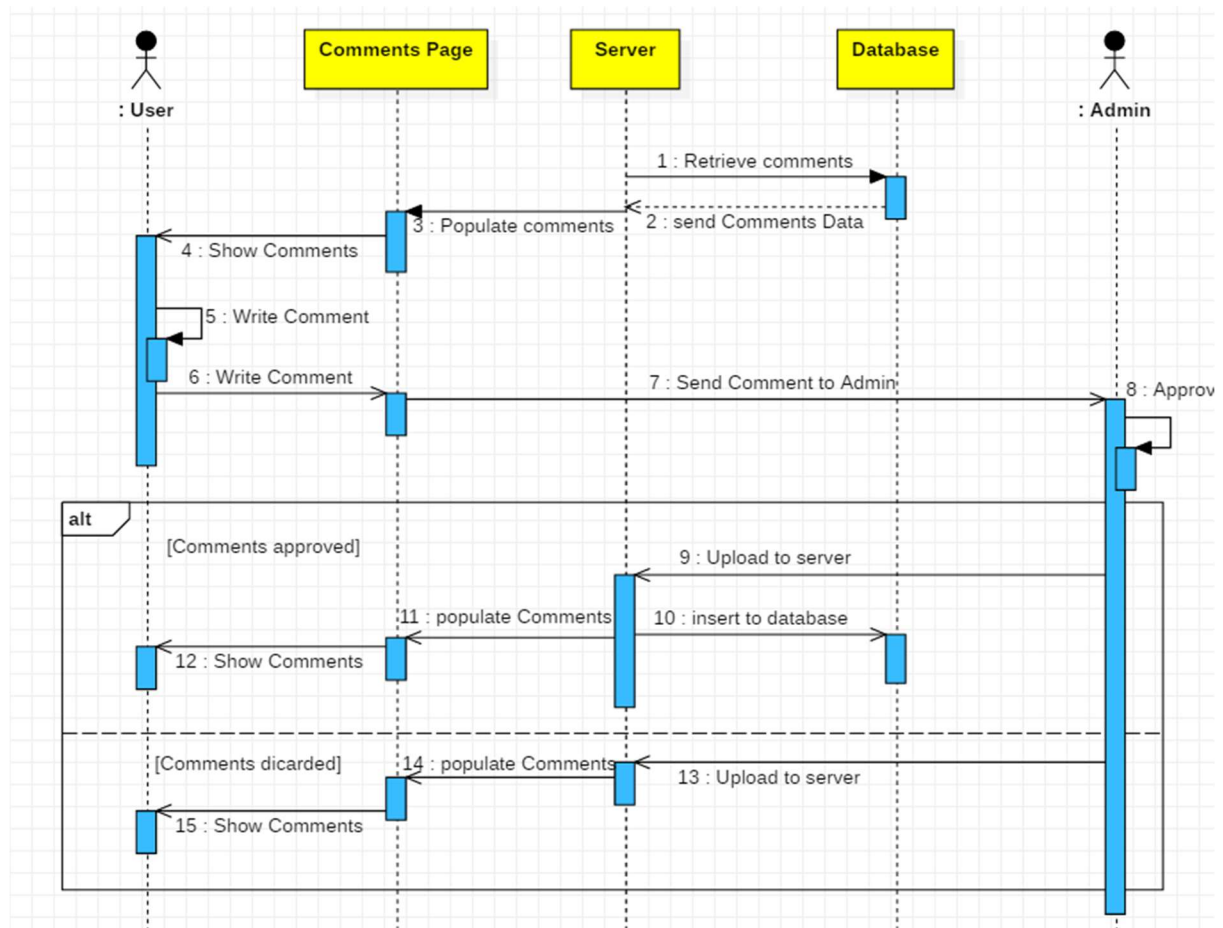


### 3. Sequence Diagrams:

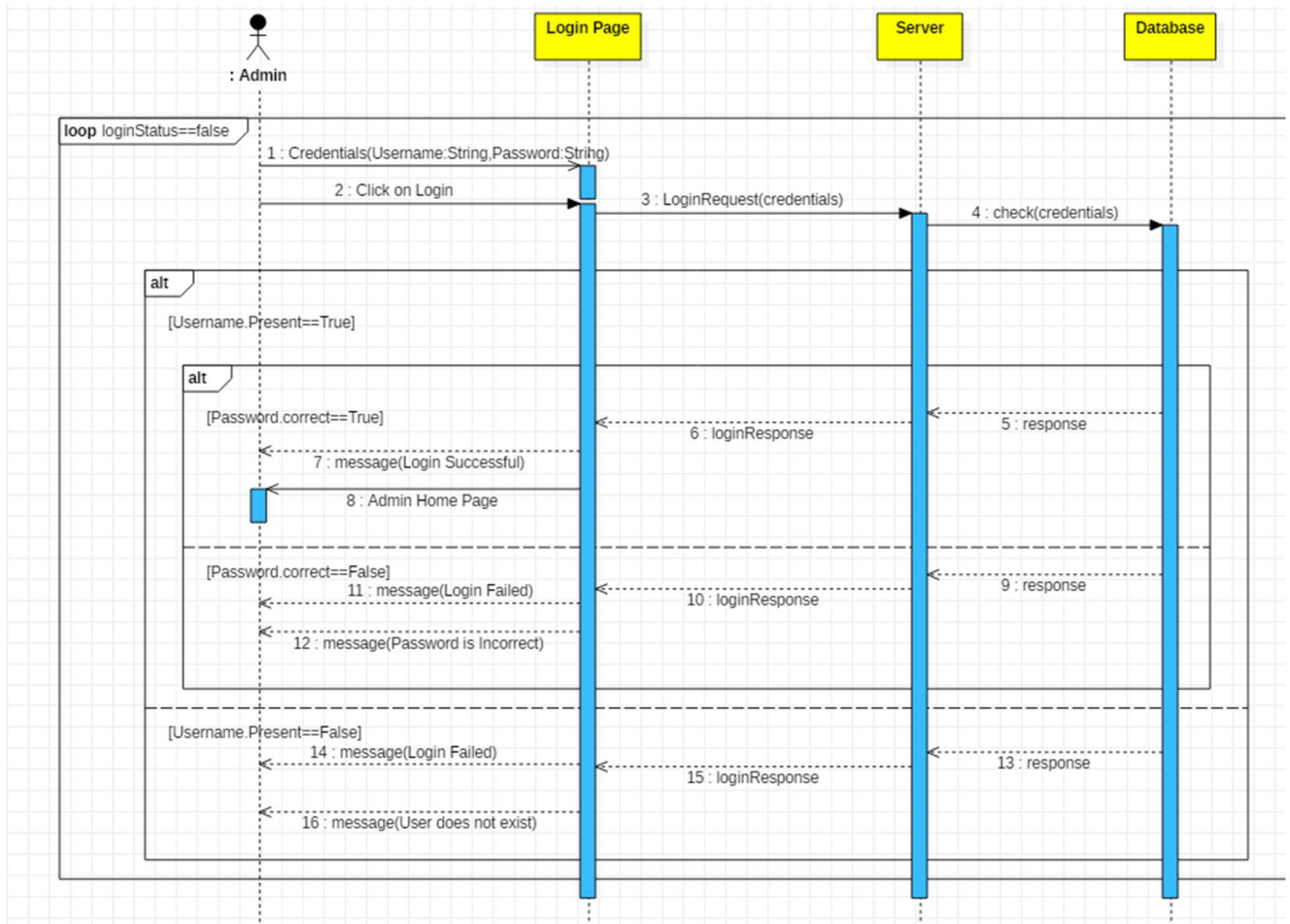
#### 3.1 Access website:



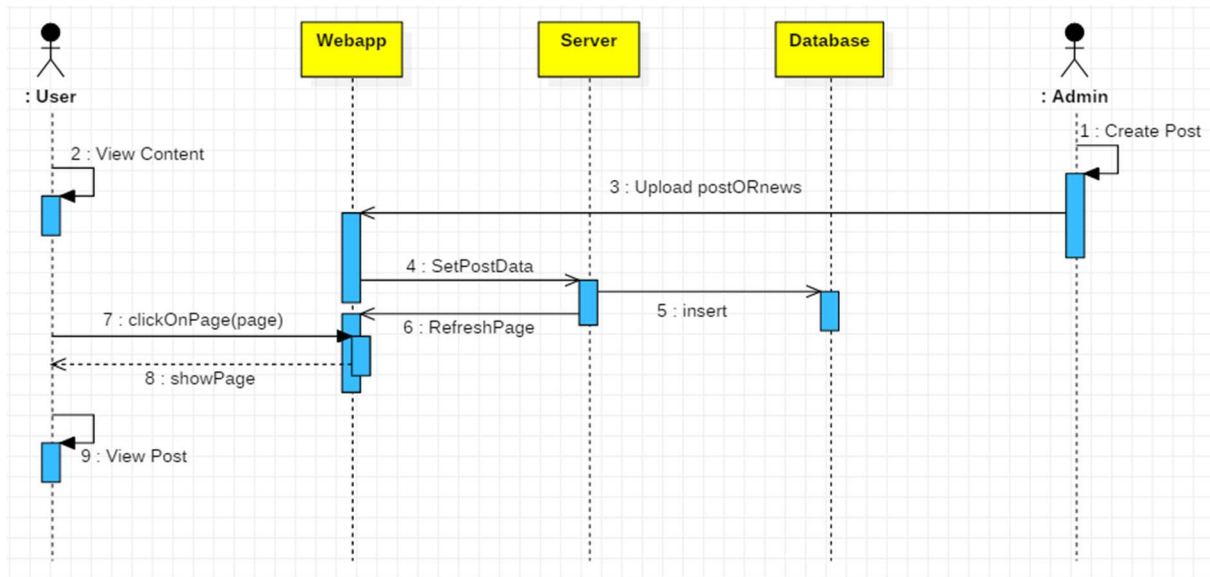
### 3.2 Comments validation:



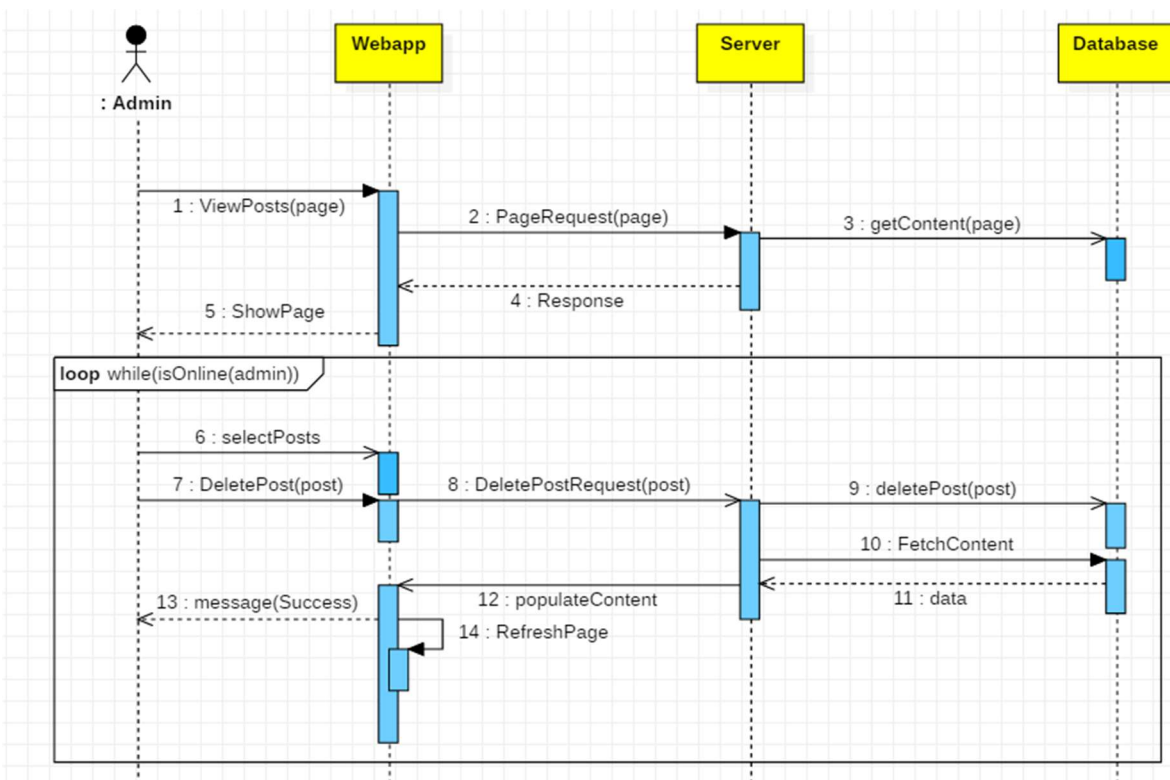
### 3.3 Admin login:



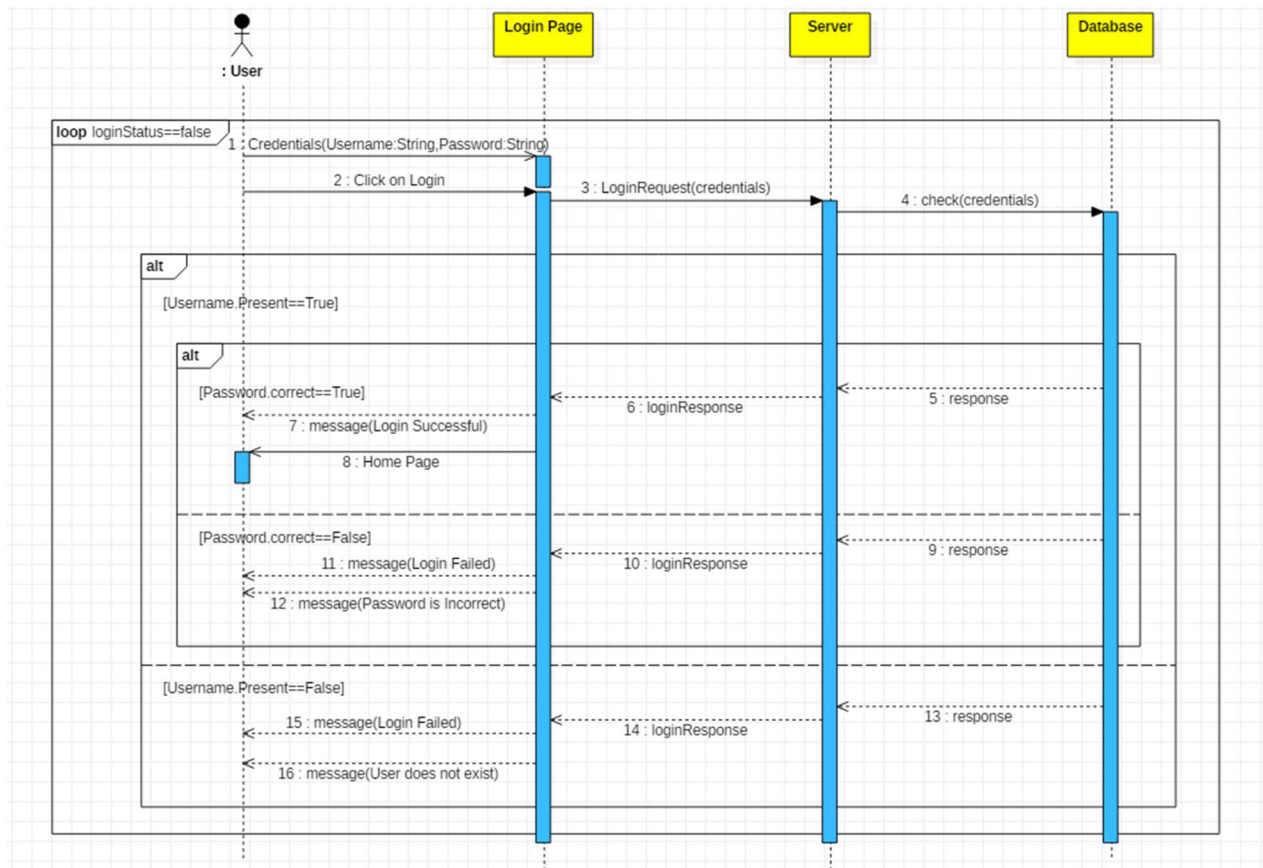
### 3.4 Create Post:



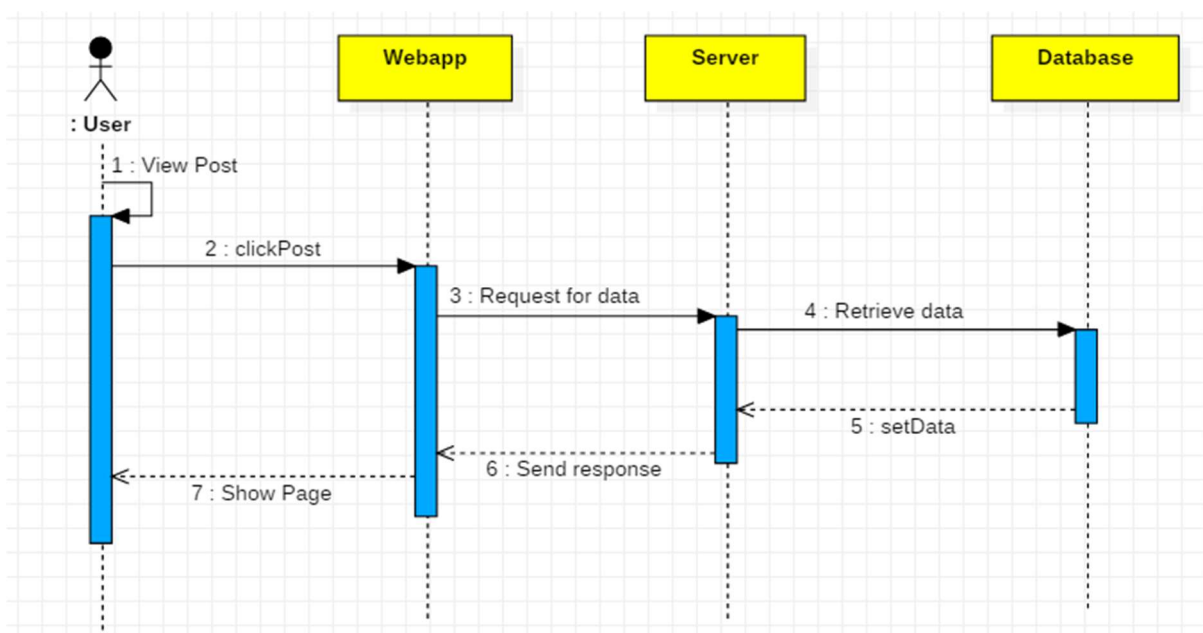
### 3.5 Delete post:



### 3.6 User Login:



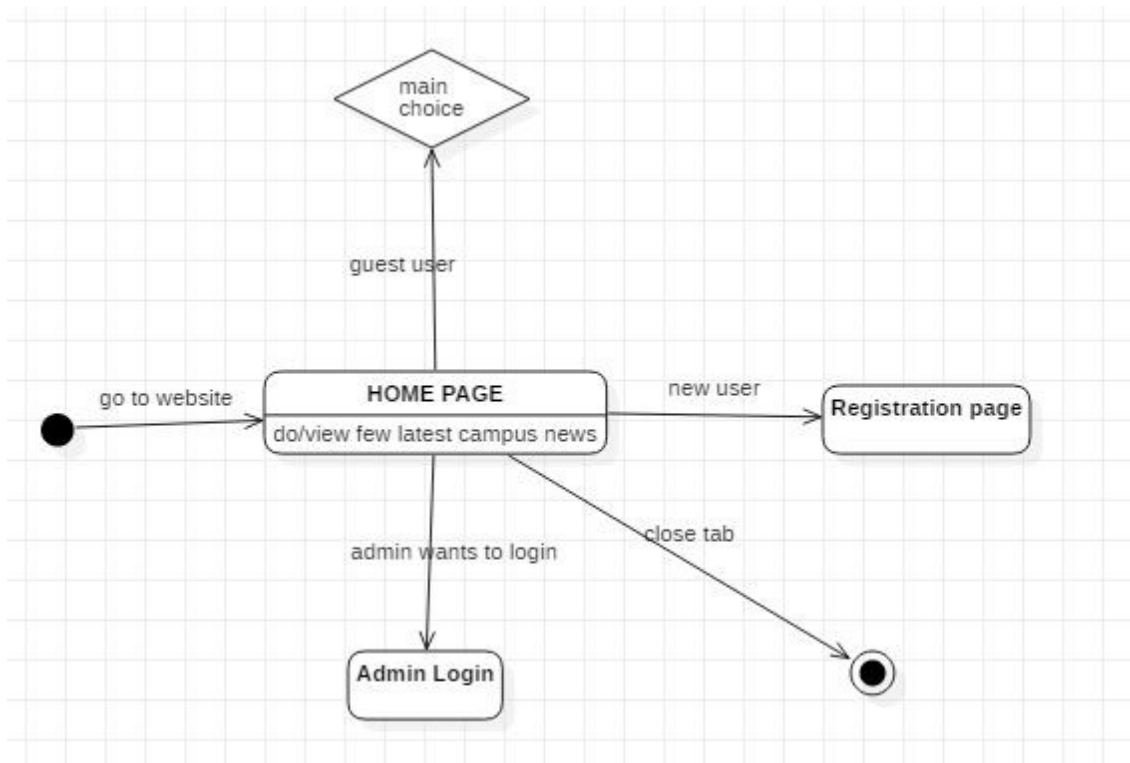
### 3.7 View Posts:



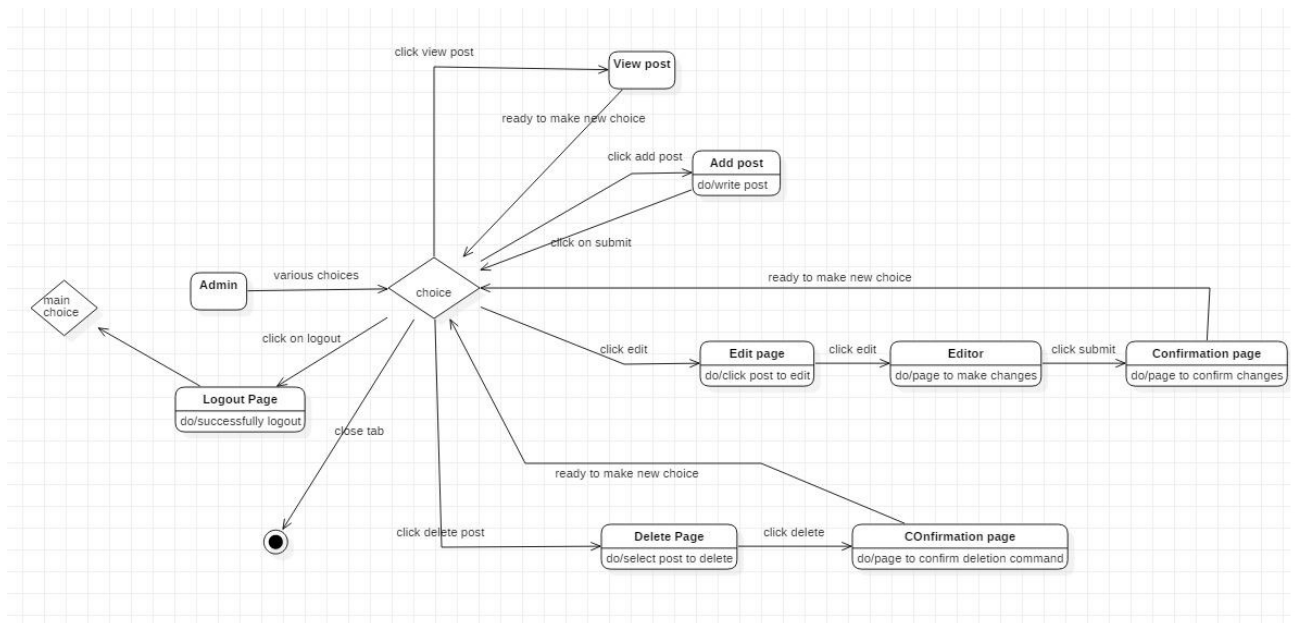
#### 4. State Diagram: Overview



### State Diagram: Home

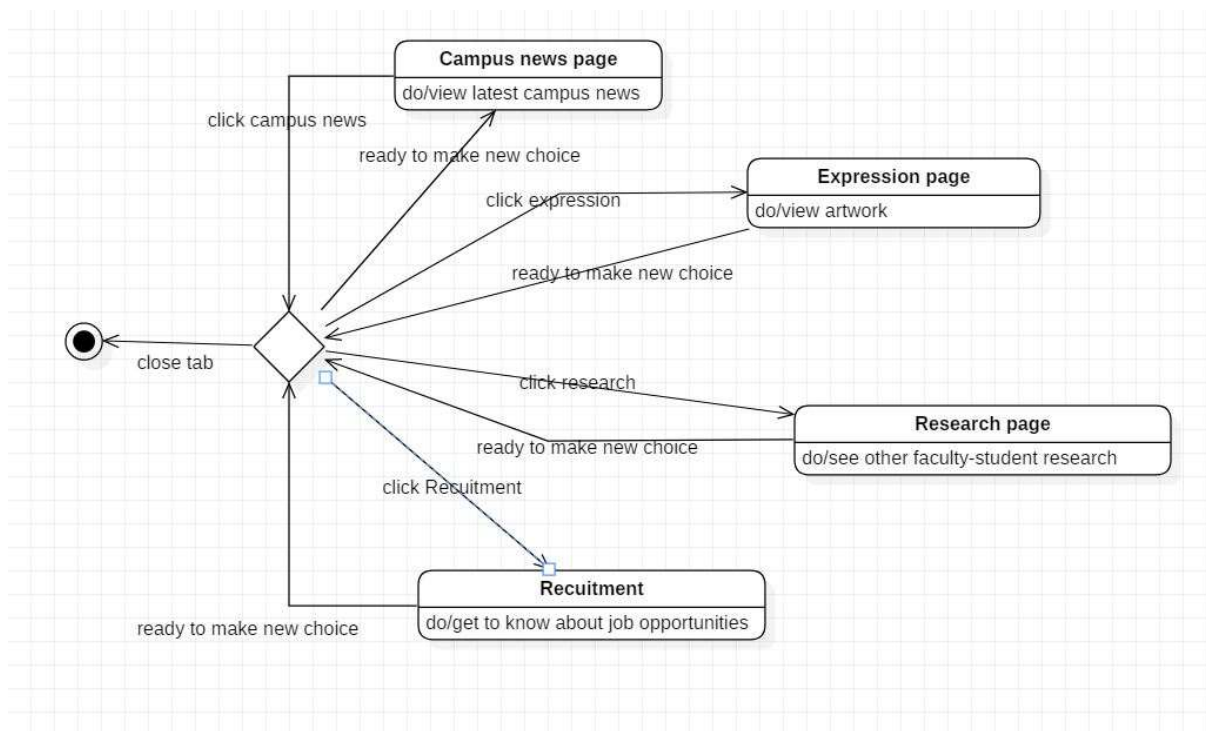
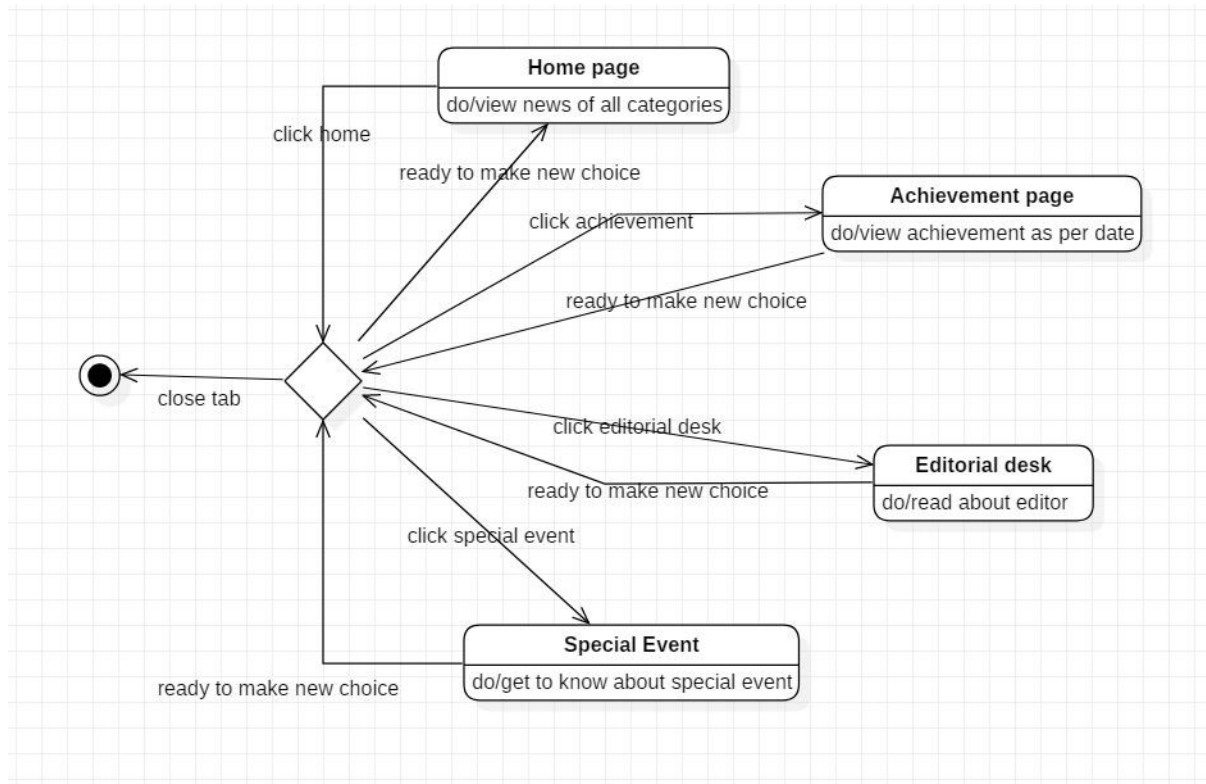


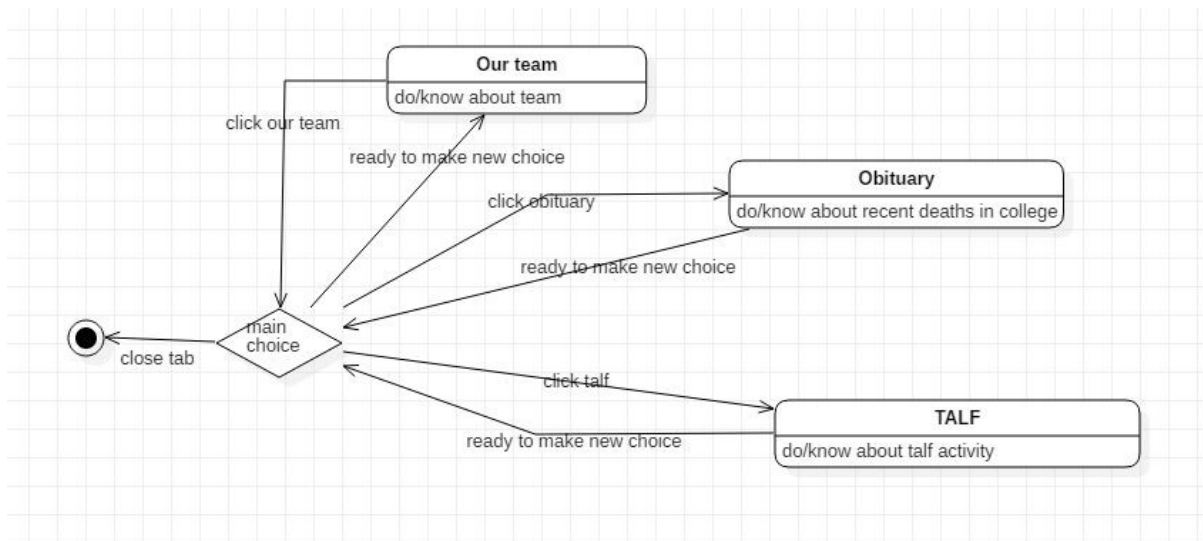
### State Diagram: Admin



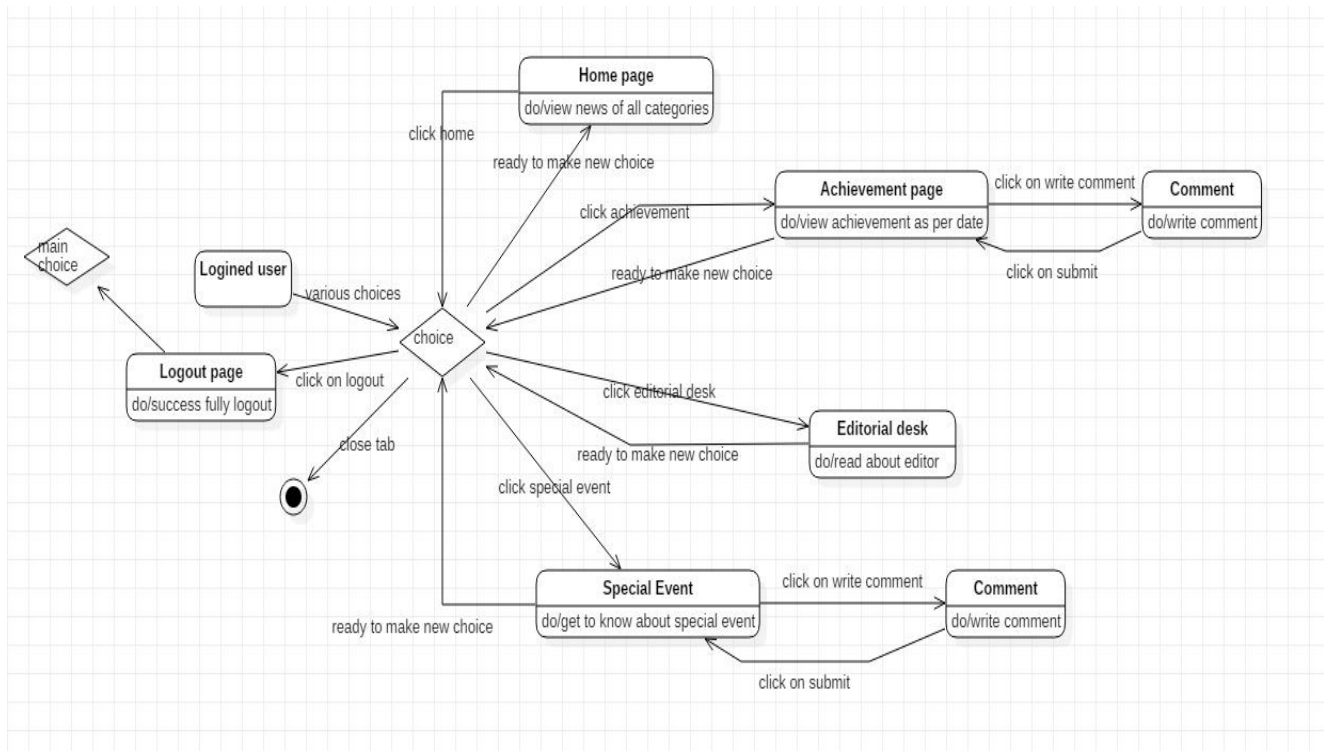


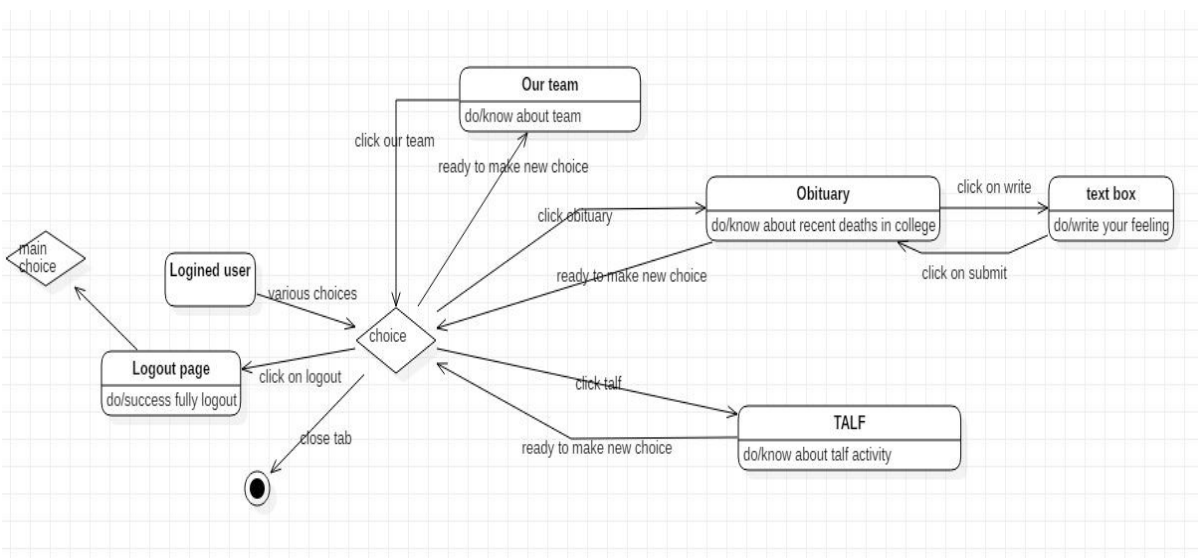
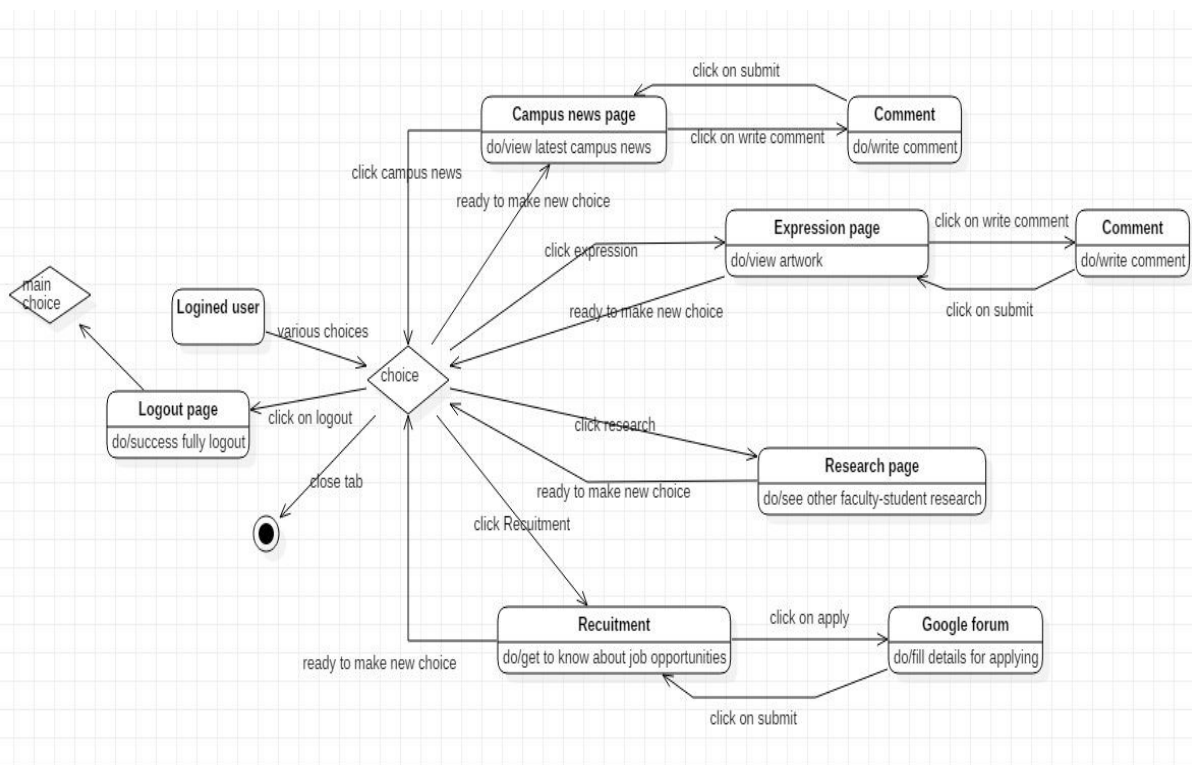
## State Diagram: Guest User



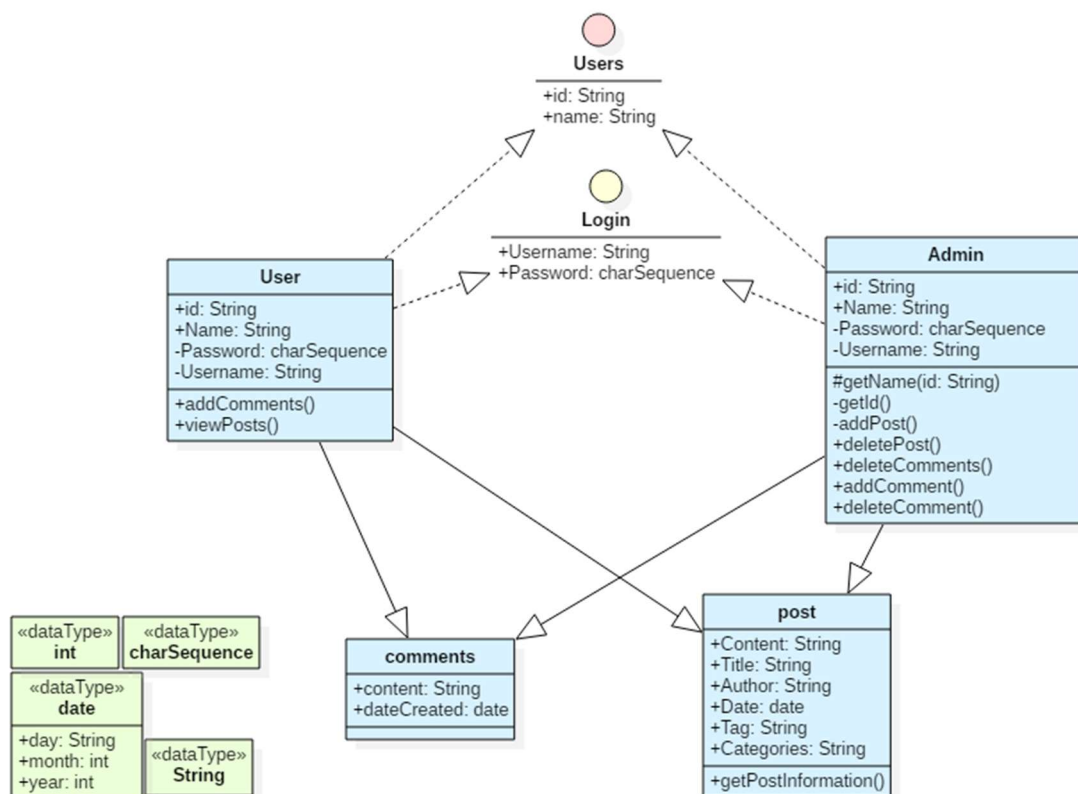


### State Diagram: Registered User





### Class Diagram:



## **5. Logical Architecture Description**

The Architecture is based on Client – Server communication where the data is stored in MySQL Database and the server retrieves the data as and when the client requires it to.

## **6. Sequence Diagrams**

### **6.1 Access website**

The User first enter the domain URL and hit enter to send request to access the page to the server. After the server has received the request it will redirect the user to the Home page of the website.

### **6.2 Comment Validation**

User can go through the news articles and can put their comments. The comments then will be evaluated by the admin. After evaluation the comments will be published.

### **6.3 Create posts**

Admin can create posts after login and upload the content to the server after which the content is published. The post may contain news articles, media files or images etc.

### **6.4 Delete Posts**

The admin can delete posts or make necessary updates if required. The updated data gets fetched by the server and is then published.

### **6.5 User Login**

The User fill the credentials for logging in to the website. By logging in the user can put their comments as the comment require a username and valid authentication.

### **6.6 View Post**

Once the user is redirected to the home page, user can view the page of choice.

### **6.7 Admin Login**

The Admin fill the credentials to log in to the website. After logging in, the admin is redirected to the admin home page and can Create, Read, Update, Delete operations on the desired posts and comments.

## 7.States Diagram:

### 7.1 Home

The User first go to the website either by logging in (Registered user) or without logging in (Guest user). Admin also require an authorized login to access the features. Once the user gets to the home page, user can read the posts and news.

### 7.2 Admin

The admin must log in first to access the functionalities. The admin gets the right to add and delete posts, read, validate, modify and delete comments.

### 7.3 Guest User

The user who only wants to view the content of the posts falls in this category. The guest user does not have the right to comment on the posts.

### 7.4 Registered User

The registered user can login to view the content and write comments. User can register to the website by clicking on the signup button and filling the required fields of the form.

## 8.Pseudo Code

### METHODS:

#### 1.LOGIN (String user)

```
I.  if (username=valid &&
      password==valid)
      a.  if (user==Admin)
           then, go to Admin desk
      b.  if (user == users)
           then, go to user desk
      c.  else go to 1.
II. else go to 1.
```

## 2.VIEW\_PAGE (String user)

```
I.  if (users!=NULL)
    • if (user == Admin)
        then, show
        homepage with Admin bar
    • if (user == users)
        then, show user's
        homepage
    ii. else
        show default
        homepage.
```

## 3.ADD\_COMMENTS ()

```
I.  go to comments section of the
    page
II.  write comments
III. click on postComments()
    • if ,user logged in
      as users
      then, post
      comments
    • if ,user logged in
      as Admin
      then, go to
      comment_verification
      ();
      else
      go to
      comment_verification
      ();

IV.  page gets refreshed.
```

#### 4. COMMENT\_VERIFICATION ()

```
i. Requires login(Admin)
    1) if
        (comments==valid)
            then, approve the
               comments
    2) else
        delete_comments ();
ii. comments page
```

#### 5. DELETE\_COMMENTS ()

```
a) Requires login(Admin)
b) Go to comments section in the
   dashboard
c) If (comment==invalid)
    then, remove(comment);
    d) else
        Go to comments_verification().
```



## 6. ADD\_CATEGORIES ()

- a) Requires login(Admin)
- b) Click on add categories section
- c) If (category!=NULL)  
then, Add category  
else  
Intent to desk-board

## 7. ADD\_POST ()

- a) login (user);
- b) Go to all post
- c) Go to new post
  - i. Add title
  - ii. Select category
  - iii. Add content
    - If media present
      - Upload from the local drive
    - else
      - Continue with the post
  - iv. Add the thumbnail picture
- d) Upload the post.
- e) Check the page for the change

## 8. UPDATE\_POST ()

- a. Requires login(Admin)
- b. Go to all post
- c. Search the post need to be updated
- d. Update the change.
- e. Save changes

## 9. DELETE\_POST()

1. Requires login(Admin)
- 2.Go to all post
3. Click on search bar
- 4.type the post name to be  
deleted
- 5.click on delete
- 6.Save the changes

## 10. UPDATE\_CATEGORIES ()

- a. Requires login (Admin)
- b. Go to categories
- c. Search the category need to be updated
- d. Update the changes
- e. Save changes

## 11. ADD\_USER ()

1. Requires login(Admin)
2. go to all user
3. go to new user and fill all the details given below:
  - a. User name:
  - b. email:
  - c. Name:
  - d. password
  - e. set the role
4. Click on add new user

## 12. DELETE\_USER ()

1. Requires login(Admin)
2. go to all user
3. select the user you want to delete
- 4.click delete user
- Delete all content  
    Attribute
- all content to
5. Choose either according to your need
- 6.Confirm deletion.

## 13. UPDATE\_USERDETAILS (String user)

1. Requires login (Admin)
- 2.if (user==Admin)  
    Change the user details you want to change
- Go to send a copy to the user.

```
3.if(user==users)
```

```
    Go to dashboard
```

```
                                -> change the  
necessary details.
```

```
4.save the changes
```

#### **14. ADD\_PAGE ()**

```
1.Login ();
```

```
2. Go to all page
```

```
3. Go to new page
```

```
    a. Add title
```

```
    b. Add content
```

```
    c. If media present
```

```
        then,Upload from the local  
drive
```

```
    e. else
```

```
        Continue with the post  
Add the thumbnail picture
```

```
4. Upload the page.
```

```
5.Check the page for the change
```

### 15. UPDATE\_PAGE ()

- a. login ()
- b. Go to all page
- c. Search the page need to be updated
- d. Update the change.
- e. Save changes

### 16. DELETE\_PAGE ()

1. Login ()
- 2.Go to all page
3. Search the post need to be DELETE
- 4.delete the post
- 5.Save the changes

### 17. DELETE\_CATEGORY ()

1. Login ()
- 2.Go to all category
3. Search the category need to be  
DELETE
- 4.delete the category
- 5.Save the changes

\*\*\*\*\*