# 

COURSE LEARNING WEBSITE

ANALYSIS DOCUMENT

CEN 4010 - U01

SPRING 2016

By:

Daylin Gonzalez

Nicolas Perez

Dhrumel Shah

INTRODUCTION

The internet today is being used by almost half the world’s population for a variety of purposes. Over the last decade, society has used technology to revolutionize the education system and many students now opt for online degrees and certification programs for cheaper and more convenient ways of learning. Even students who attend college lectures benefit from online resources to build on their knowledge. Our project aims at creating a course learning website that can assist anyone interested in learning college courses. It provides students with easy access to video tutorials, articles, animations and other resources relevant to specific courses. It also contains discussion boards in order to create a shared and complete learning experience for the users.

OVERVIEW

This course learning website will be developed using an object-oriented software development approach. The website contains a list of courses and each course has its own page for course materials and a discussion board. Users can access and upload course materials and also post on the discussion board.

A course material also has tags so that the user can search for course materials by entering the course name or a tag name. When we say that we *tag* something it is the same as marking it with a name or keyword so that it can be found by this name or keyword by the user.

Each course material also has a user rating, and the materials for each course are sorted by each user’s preferences. A user can add courses to his/her favorite list of courses so it can be viewed easily once the user accesses his/her account.

The website allows access to 3 types of users: Guest, Member and Admin.

Functional requirements

**Guest**

1. ~~Create Account~~
2. ~~Search for course materials~~
3. ~~Login~~
4. View course materials
5. ~~View discussion board~~

**Member**

All privileges of a guest plus the following:

1. ~~Add course material~~
2. ~~Add tag to course material~~
3. ~~Rate course material~~
4. ~~Add discussion post~~
5. ~~Add course to favorites~~
6. ~~View favorite courses~~
7. ~~Remove course from favorites~~
8. ~~Delete account~~
9. ~~Logout~~

**Admin**

All privileges of a member plus the following:

1. Delete member account
2. ~~Add course~~
3. ~~Delete course~~
4. Remove tags
5. ~~Delete course material~~
6. ~~Delete discussion post~~

Non-functional requirements

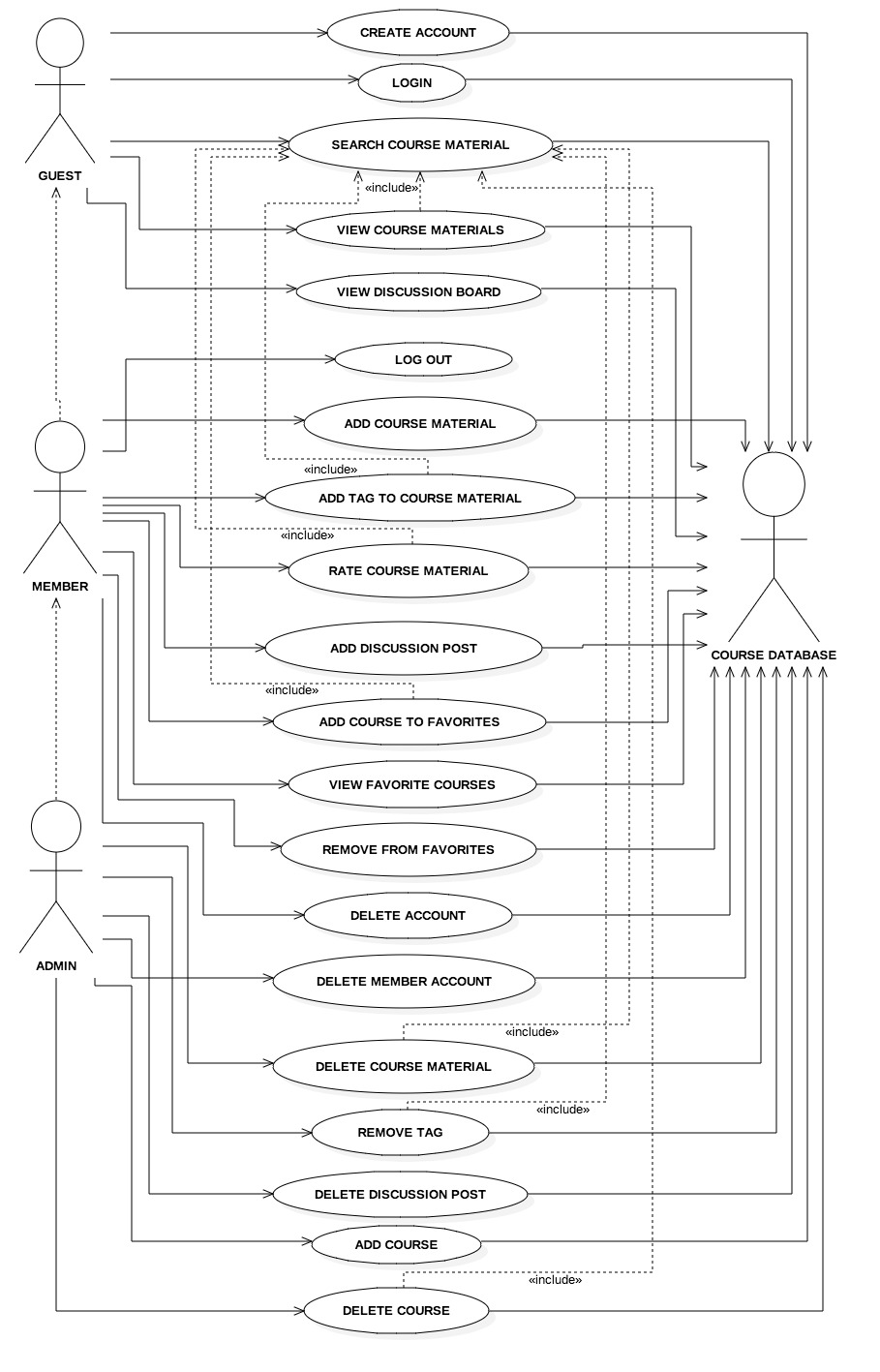
* The system will have three types of users (guest, member, and admin)
* System will be available through the web.
* The system should be available through a URL endpoint.
* The system response time should be acceptable for the user. Acceptable means that the response time does not exceed more than 5 seconds unless exceptional cases caused by external entities to the system such as the loss of power and/or connectivity.
* The user interface can experiment graphical changes such as the redesign of the HTML content or any other presentation layer elements, which implies that the system should be built in a way that these changes do not affect other layers of system during these changes.
* The system should be running on a cloud hosting environment.
  + backups are automatically done by the cloud hosting.
  + Monitoring and instrumentation of system is done by the cloud hosting.
* A continuous integration (CI) environment will deploy our system to the cloud hosting. A CI system allows us to automatically deploy new changes to our production environment reducing the manual, tedious, and prone to errors process of manual deployments. The deployments are automated by the CI tool.

Constraints (“Pseudo requirements”)

* The implementation language must be C# on the .NET framework.
* The cloud based hosting on Windows Azure.

System models

1. Use Case Model - Use Case Diagram and Use Cases



**1) Create Account**

This Use Case describes the process by which a user can create an account and become a Member.

**Actor**

* Guest

### **Flow of Events**

1. The Use Case starts when the user clicks on the Create Account button in the main page.
2. The system displays a form for the guest to enter information.
3. The user enters the following information:
   * + - 1. User name
         2. Email Address
         3. Password
4. The user clicks on the Submit button.
5. The account details are stored in the database.
6. The user is logged in.

**2) Search Course Material**

This Use Case describes the process by which a user can search for a course material.

**Actor**

* Guest
* Member
* Admin

### **Flow of Events**

1. The Use Case starts when the user clicks on the search bar in the main page
2. The user enters the name of a course or a tag.
3. The user clicks the Search button.
4. The system retrieves course materials of the selected course or tag from the database.
5. The system displays course materials on the course page.

**3) Log In**

This Use Case describes the process by which a user can log in.

**Actor**

* Guest

### **Flow of Events**

1. The Use Case starts when the user clicks on the Login button in the main page.
2. The system shows the Login page.
3. The user enters his/her Username and Password.
4. The user clicks the Submit button.
5. The system verifies the user’s login credentials in the database.
6. The user is logged in if the details are found in the database.
7. The user (who is now a member) is redirected to the course page.

Exceptions:

1. If user details are not found, the system displays a message indicating that the login has failed.
2. The user is redirected to the application main page.

**4) View Course Materials**

This Use Case describes the process by which a user can view a course material.

**Actor**

* Guest
* Member
* Admin

### **Flow of Events**

1. The Use Case starts when the user searches a course material.
2. The user clicks on the course material.
3. The system opens the course material.

**5) View Discussion Board**

This Use Case describes the process by which a user can view the discussion board

**Actor**

* Guest
* Member
* Admin

### **Flow of Events**

1. The Use Case starts when the user searches a course material by entering a course name.
2. The system navigates to course page.
3. The course page displays the discussion board.

**6) Add Course Material**

This Use Case describes the process by which a user can add a course material.

### **Actor**

* Member
* Admin

### **Flow of Events**

1. The Use Case starts when the user searches for the course material by entering the course name on the main page.
2. The Use Case starts when the user clicks on the Add button on a specific course page.
3. The system displays a form that includes the following information:
   1. *course material name*
   2. *links*
   3. *text*
   4. *attachment documents box*
4. The user fills out the information and adds the content.
5. The system also displays a tag form so the user can *tag* the new material.
   1. The user can select *tags* from the existing tag list.
   2. The user can create new tags by typing the tag name.
6. The user clicks on the Confirm button to confirm the content.
7. The system adds the new content in the course page and stores it in the database.

**7) Add Tag to Course Material**

This use case describes the process by which a user can add tags to an existing course material.

### **Actor**

* Member
* Admin

### **Flow of Events**

1. *Use Case 2* is executed (Entry condition)
2. Once user has executed *Use Case 2,* the user clicks on the Add tags button near the course material.
3. The system displays a tag form so the user can *tag* the new material.
   1. The user will select *tags* from existing tag list.
   2. The user can create new tag by typing the tag name.
4. The user clicks on the Confirm button.
5. The system adds the tags to the course material and the tag list in the database.

**8) Rate Course Material**

This use case describes the process by which a user can rate a course material.

### **Actor**

* Member
* Admin

### **Flow of Events**

1. *Use Case 2* is executed (Entry condition)
2. The user clicks on one of the 5 stars next to the course material to be rated.
3. The system displays the star rating made by the user.
4. The rating for the course material will be saved in the database.
5. The system will update the average rating value.

**9) Add Discussion Post**

This use case describes the process by which a user can add a post to the Discussion Board.

### **Actor**

* Member
* Admin

### **Flow of Events**

1. The Use Case starts when the user searches for the course material by entering the course name on the main page.
2. The user clicks on the Add Post button in the Discussion board on the course page.
3. The system will display a textbox.
4. The user types its post into the textbox.
5. The user click the Post button.
6. The system saves the new post in the database.
7. The system will display the post on the discussion board.

**10) Add course to favorites**

This use case describes the process by which a user can add a course to his/her favorites courses.

### **Actor**

* Member
* Admin

### **Flow of Events**

1. The Use Case starts when the user clicks the Add to Favorites button on a course page.
2. The system saves the course in the user’s favorite courses in the database.
3. The system Shows a confirmation message

**11) View favorite courses**

This use case describes the process by which a user can view courses saved in Favorites.

### **Actor**

* Member
* Admin

### **Flow of Events**

1. The Use Case starts when the user clicks the My Account Menu item.
2. The user clicks the Favorites sub menu item.
3. The system retrieves the user’s favorite courses from the database.
4. The system displays the user’s favorite courses.

**12) Remove course from favorites**

This use case describes the process by which a user can remove a course from his or her favorites list.

### **Actor**

* Member
* Admin

### **Flow of Events**

1. The Use Case starts when the user clicks on the Remove button near the course in the Favorites sub menu item.
2. The system removes that course from the user’s favorites list.

**13) Delete account**

This use case describes the process by which a user can delete his/her account.

### **Actor**

* Member
* Admin

### **Flow of Events**

1. The Use Case starts when the user clicks on the My Account menu item.
2. The user clicks on the Delete Account sub menu item.
3. The system deletes the account from the database.
4. The system displays a confirmation message.
5. The user clicks on the Ok button.

**14) Delete member account**

This use case describes the process by which an admin can delete a member account.

### **Actor**

* Admin

### **Flow of Events**

1. The Use Case starts when the user clicks on the Delete Account menu item
2. The system displays a list of users.
3. The user selects the user whose account is to be deleted.
4. The user clicks the Delete button.
5. The system deletes the user account from the database.

### 

**15) Add Course**

This Use Case describes the process by which a user can add courses to the system.

### **Actor**

* Admin

### **Flow of Events**

1. The Use Case starts when the user selects the Add Course menu item on the homepage.
2. The system displays a field to enter the Course Name.
3. The user enters the Course Name.
4. The user selects Submit.
5. The system will check if the Course Name does not exist in the database.

5.1 If Course Name exists in the database, the system displays a message saying that the course already exists, and ends the use case.

1. The course is added to the database.
2. The system displays a message indicating that the course has been successfully added.

**16) Delete Course**

This Use Case describes the process by which a user can delete courses from the system.

**Actor**

* Admin

**Flow of Events**

1. The Use Case starts when the user searches for the course material by entering the course name on the main page.
2. The system navigates to the specific course page.
3. The user selects the Delete Course menu item.
4. The system displays a message asking the user to confirm the deletion.
5. The user selects Confirm.
6. The system deletes the course and all its associated course materials from the database.
7. The system displays a message indicating that the course has been successfully deleted.

**17) Delete Course Material**

This Use Case describes the process by which a user can delete a course material.

**Actors**

* Admin

**Flow of Events**

1. The Use Case starts when the user executes Use Case ‘Search Course Material’.
2. The user selects Delete on a given course material.
3. The system displays a message asking the user to confirm the deletion.
4. The user selects Confirm.
5. The system deletes the Course Material from the database.
6. The system displays a message indicating that the course material has been successfully deleted.

**18) Remove Tag**

This Use Case describes the process by which a user can remove a tag from a course material.

**Actors**

* Admin

**Flow of Events**

1. The Use Case starts when the user executes Use Case ‘Search Course Material’.
2. The user selects the tag on a given course material.
3. The user selects the Remove option displayed near the tag.
4. The system removes the tag from the database.
5. The system displays a message indicating that the tag has been successfully deleted.

**19) Delete Discussion Post**

This Use Case describes the process by which a user can delete a Discussion Post from a Discussion Board.

**Actors**

* Admin

**Flow of Events**

1. The Use Case begins when the user executes Use Case ‘View Discussion Board’.
2. The user selects Delete on a Discussion Post on the Discussion Board.
3. The system displays a message asking the user to confirm the deletion.
4. The user selects confirm.
5. The system deletes the Discussion Post from the database.
6. The system displays a message indicating that the Discussion Post has been successfully deleted.

**20) Log Out**

This Use Case describes the process by which a user can logout from the system.

**Actors**

* Admin
* Member

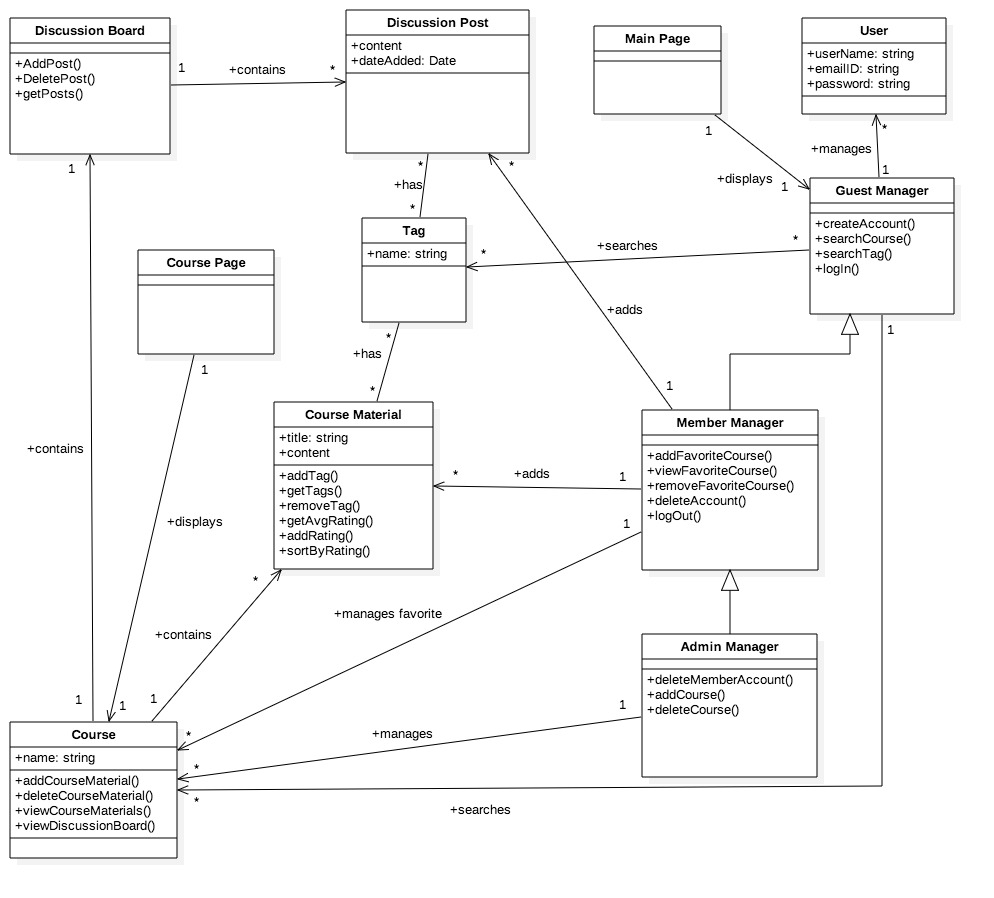
**Flow of Events**

1. The user must had executed *Use Case 3 (Login)* at some point (Required)
2. The user clicks the logout button present in any page.
3. The system present logout confirmation dialog to the user.
4. The user confirm logout.
5. The system logouts the user.
6. The application start page is shown.

Exceptions:

1. The user does not confirm the logout in the logout dialog.
2. The system hide the dialog.
3. The system show the current page (where the user was already).

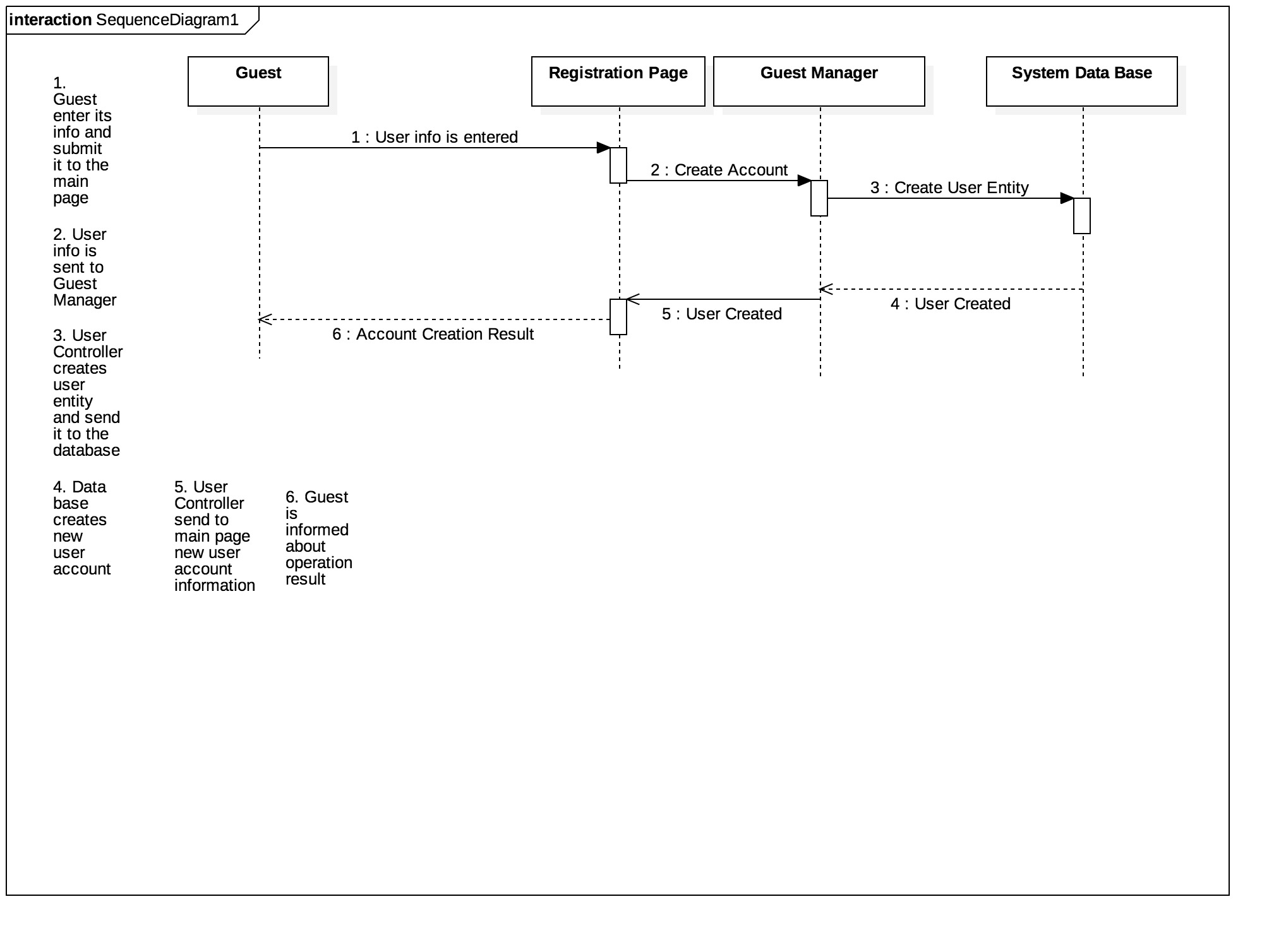
B. Object model - Class diagram



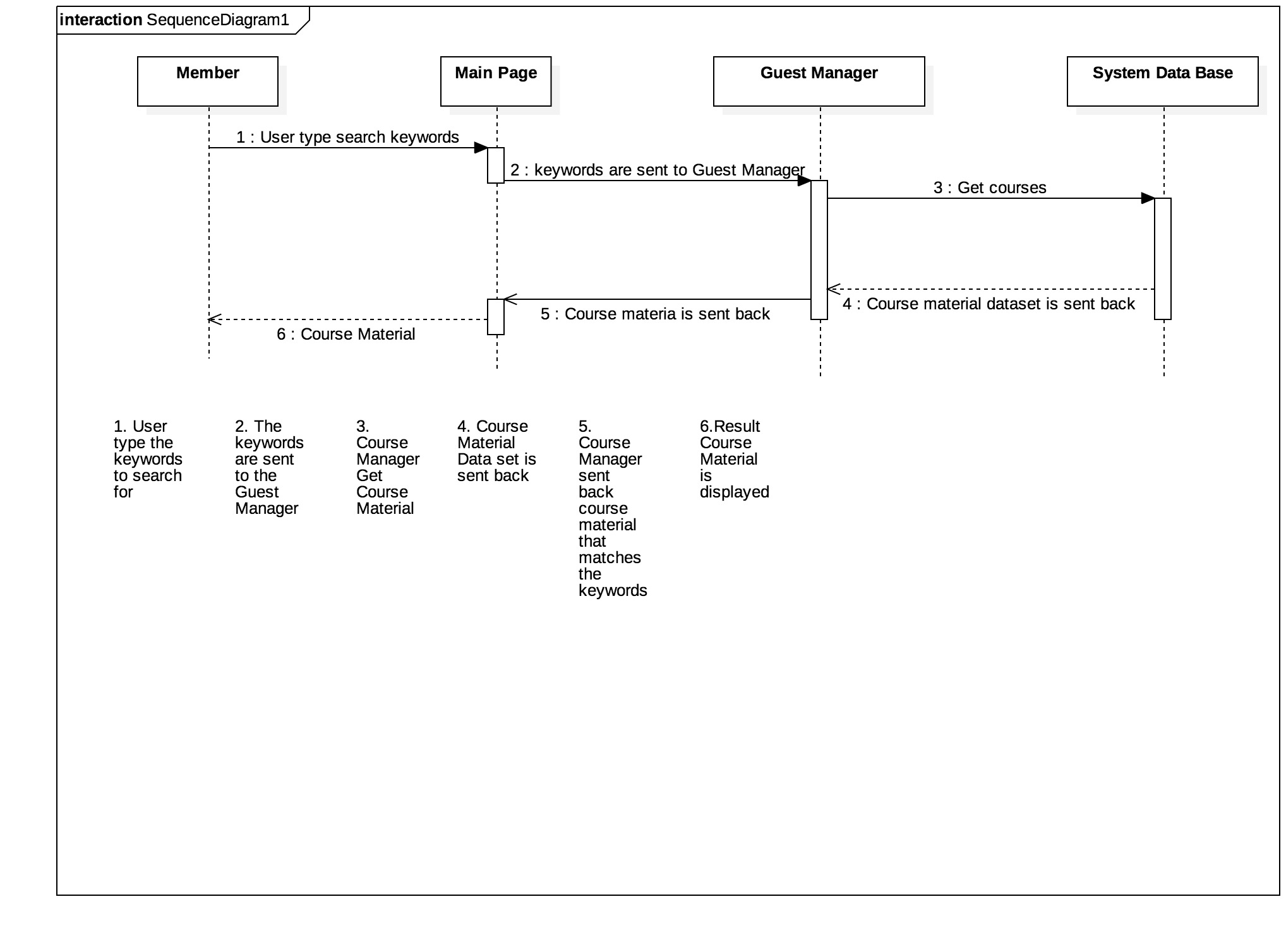
## 

C. Dynamic model - Sequence Diagrams

1. **Create Account**



**2. Search Course Material**



## 

## 

## 

## 

## 

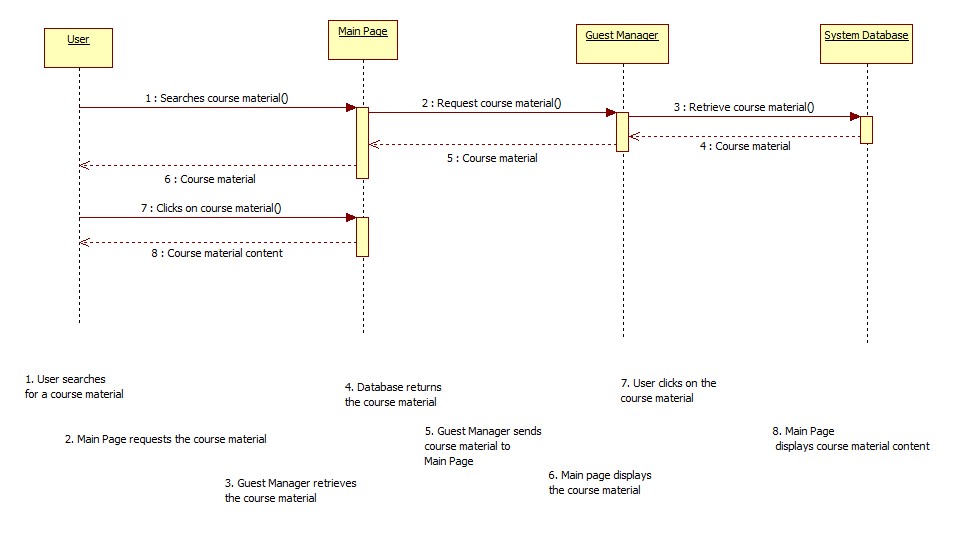
## 

## 

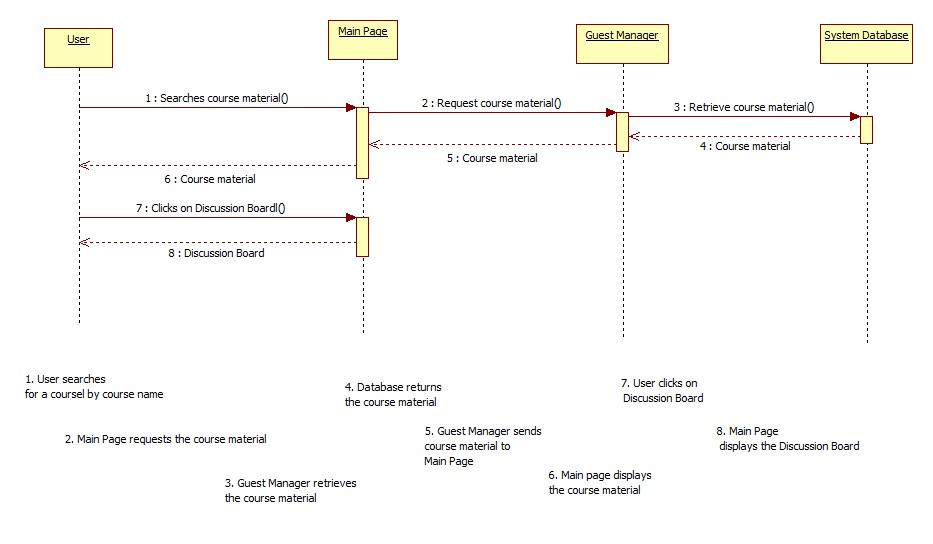
## **3. Login**

## LoginSeq.jpg

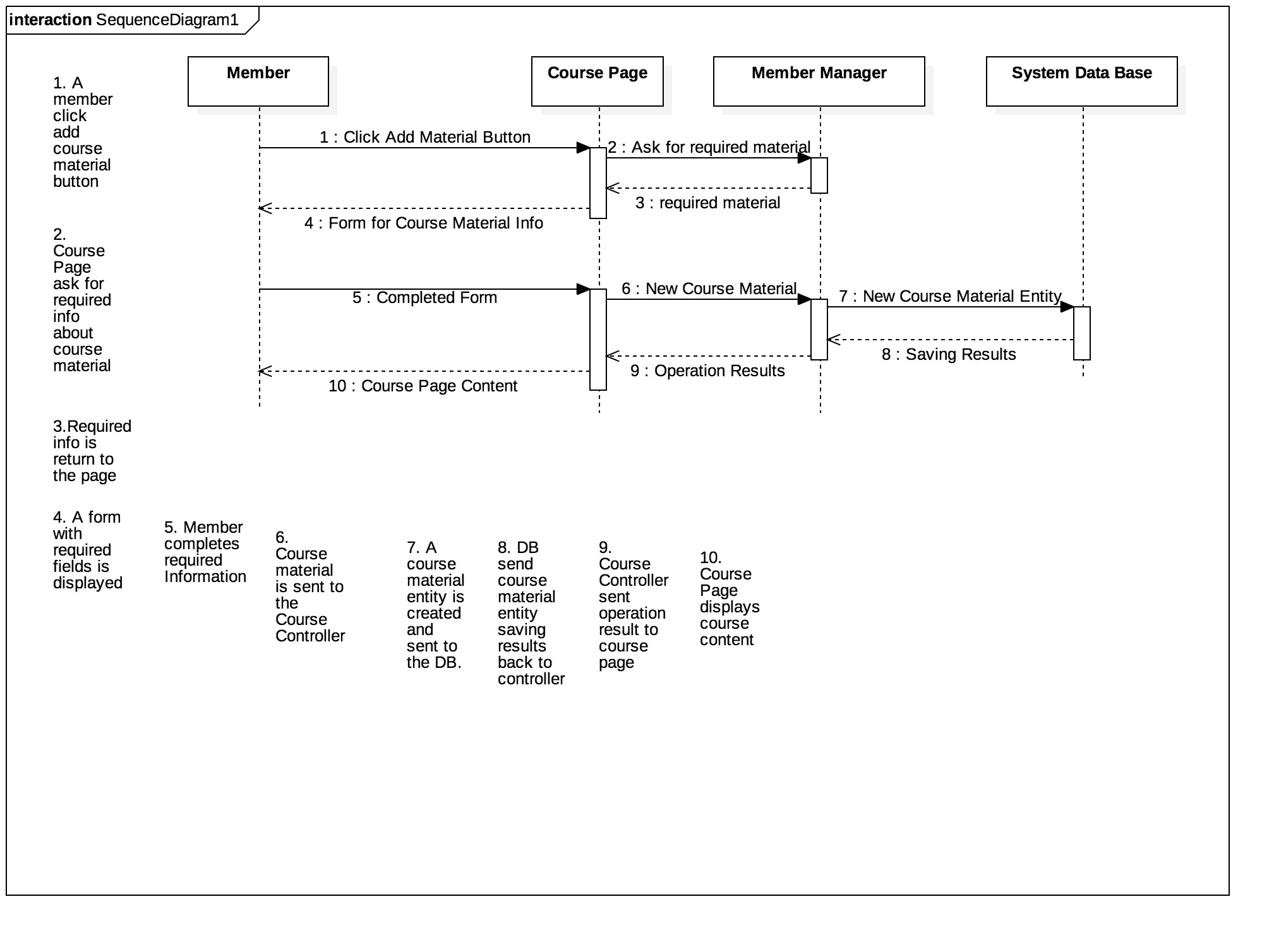
**4. View course materials**



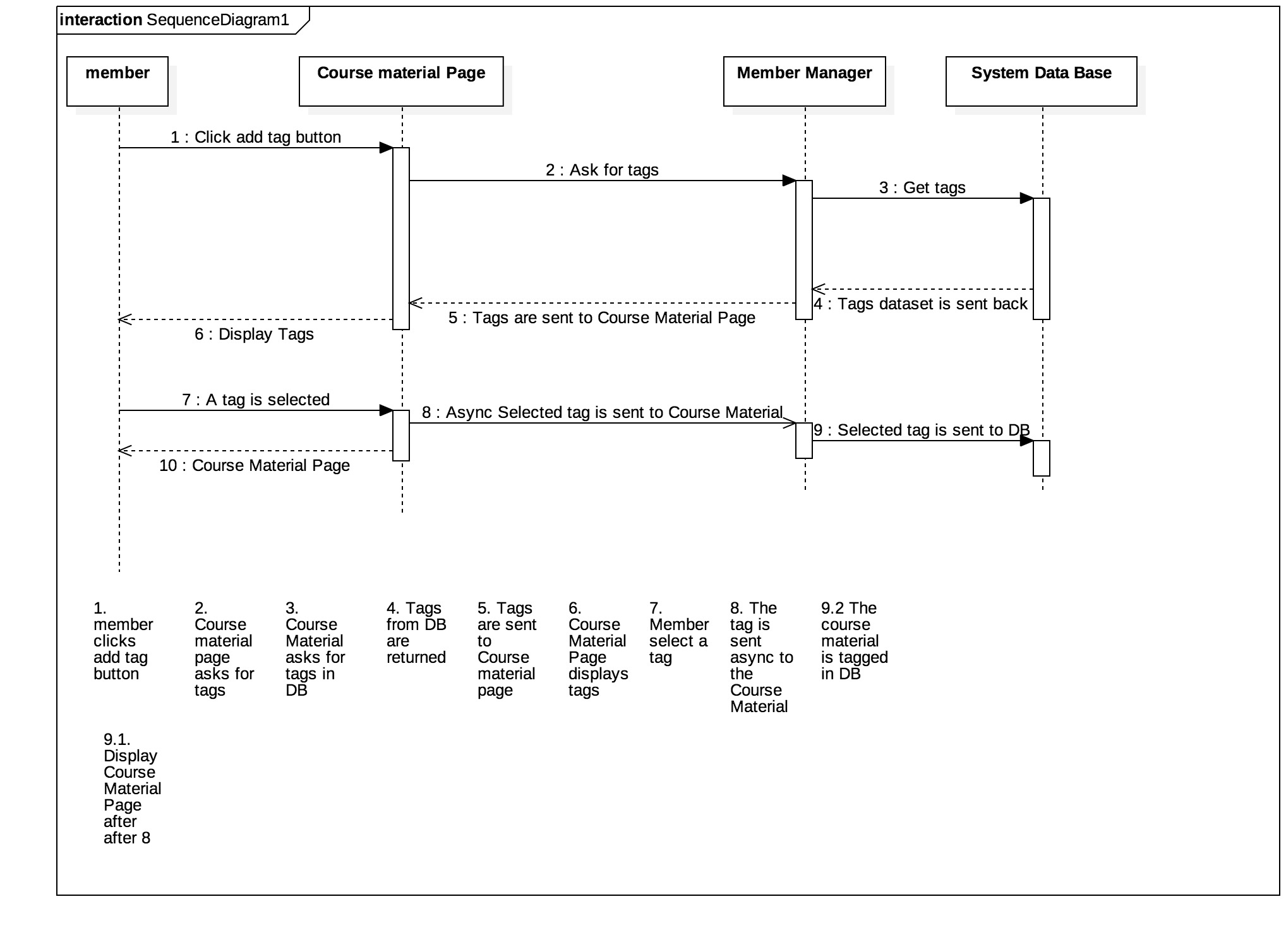
**5. View Discussion Board**

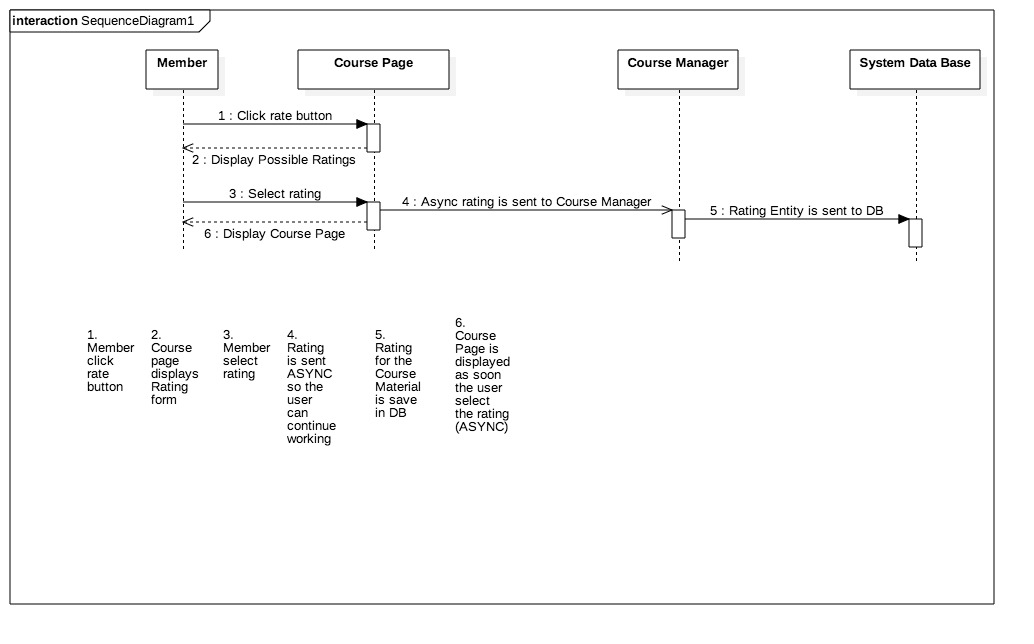


**6. Add Course Material**

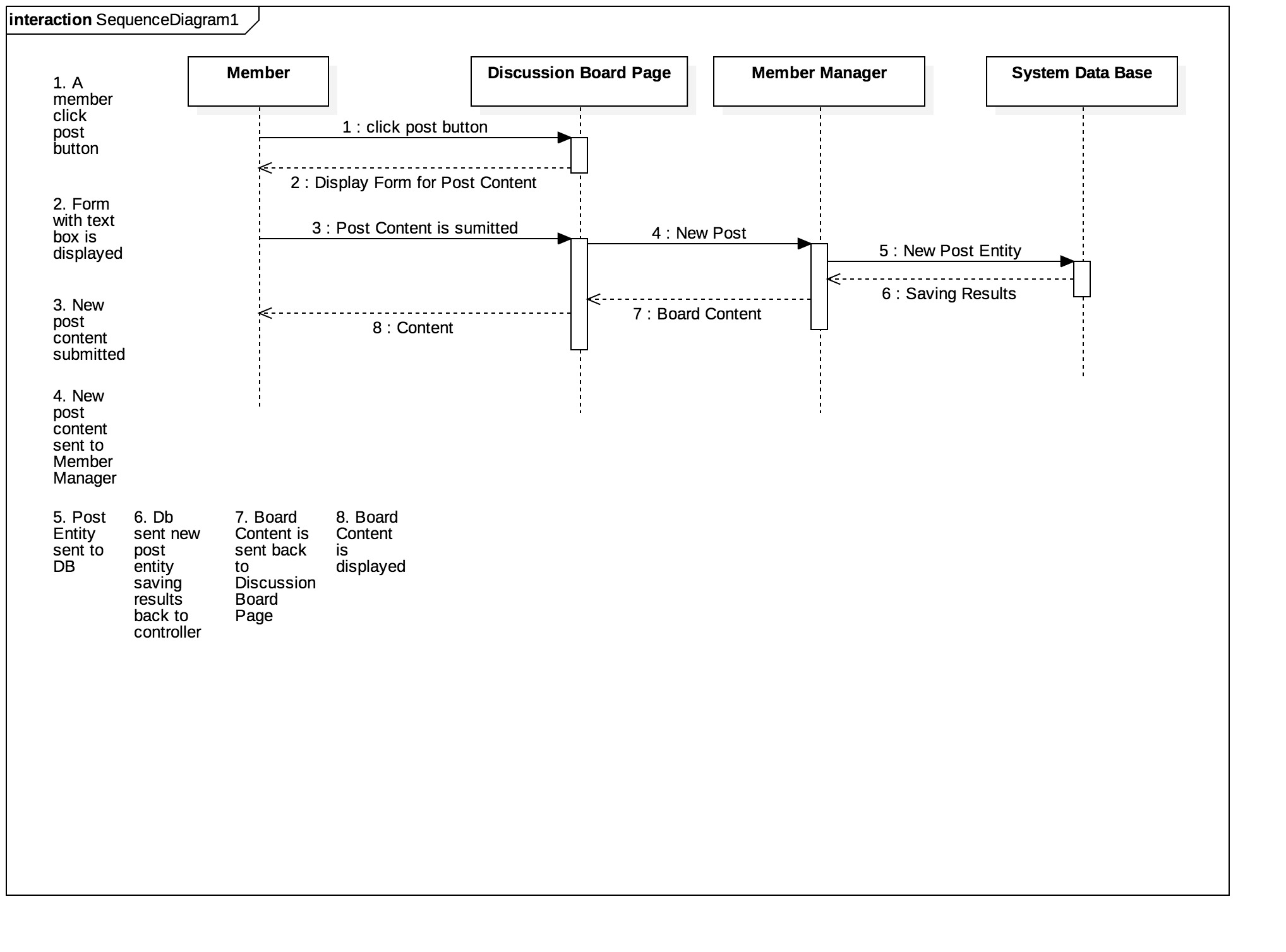


**7. Add Tag to Course Material**

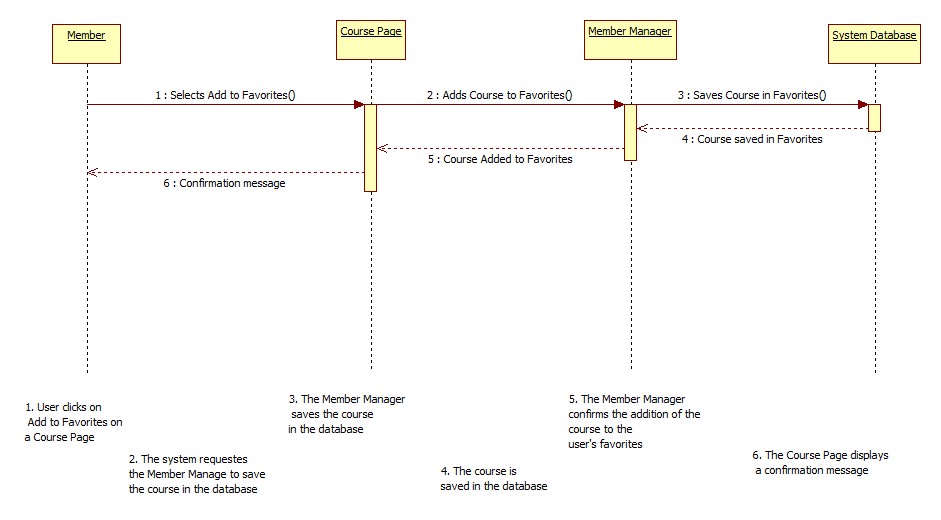


**8. Rate Course Material**

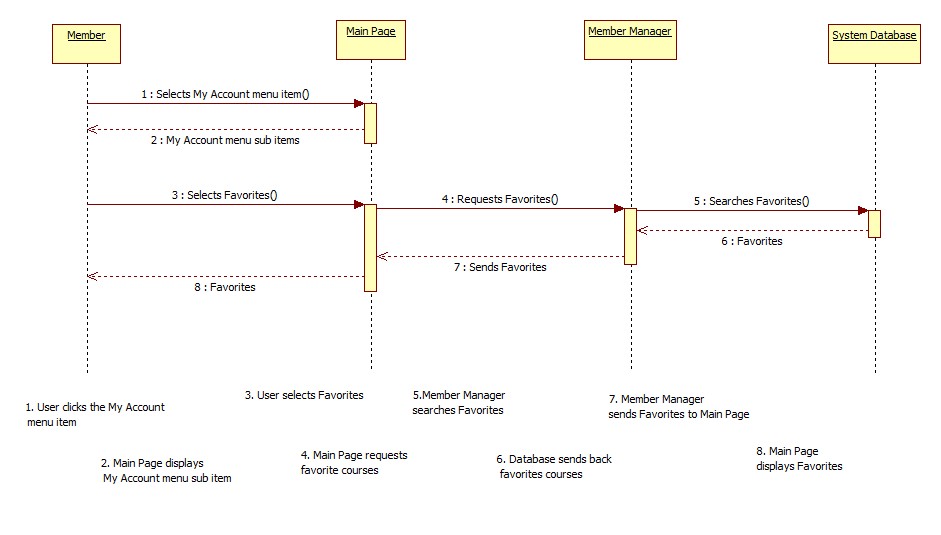
**9. Add Discussion Post**



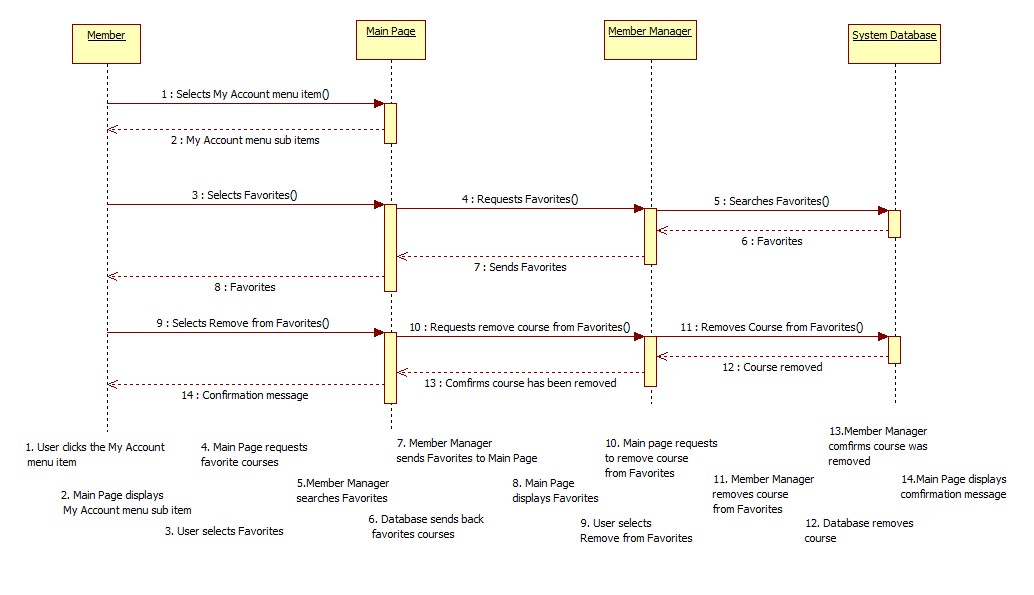
**10. Add Course to Favorites**



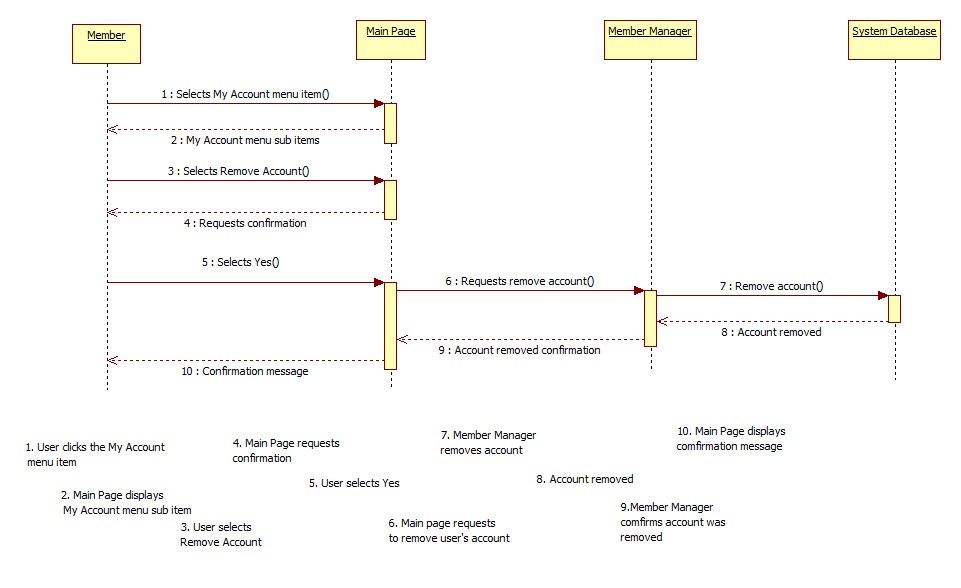
**11. View Favorite Courses**



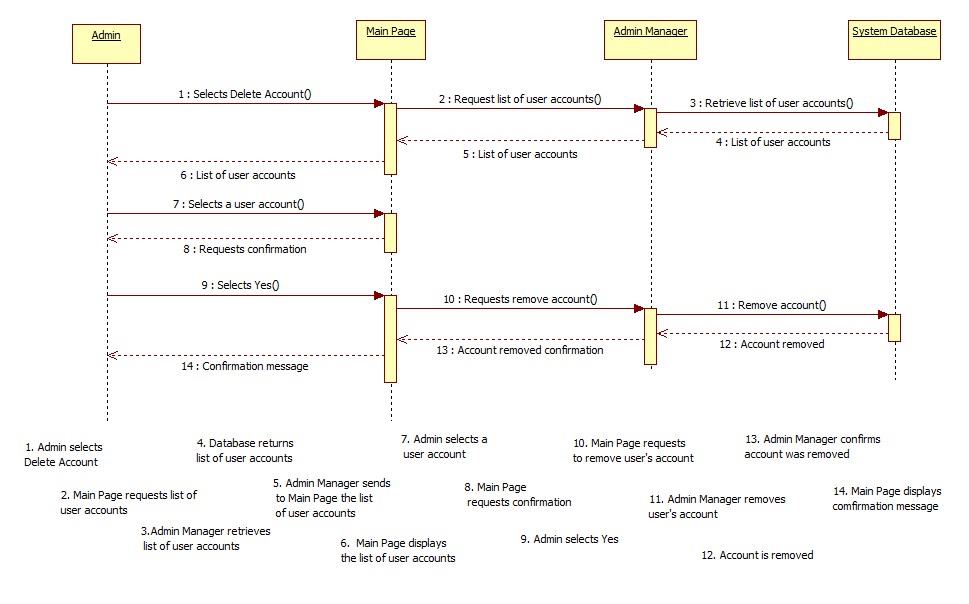
**12. Remove Course from Favorites**



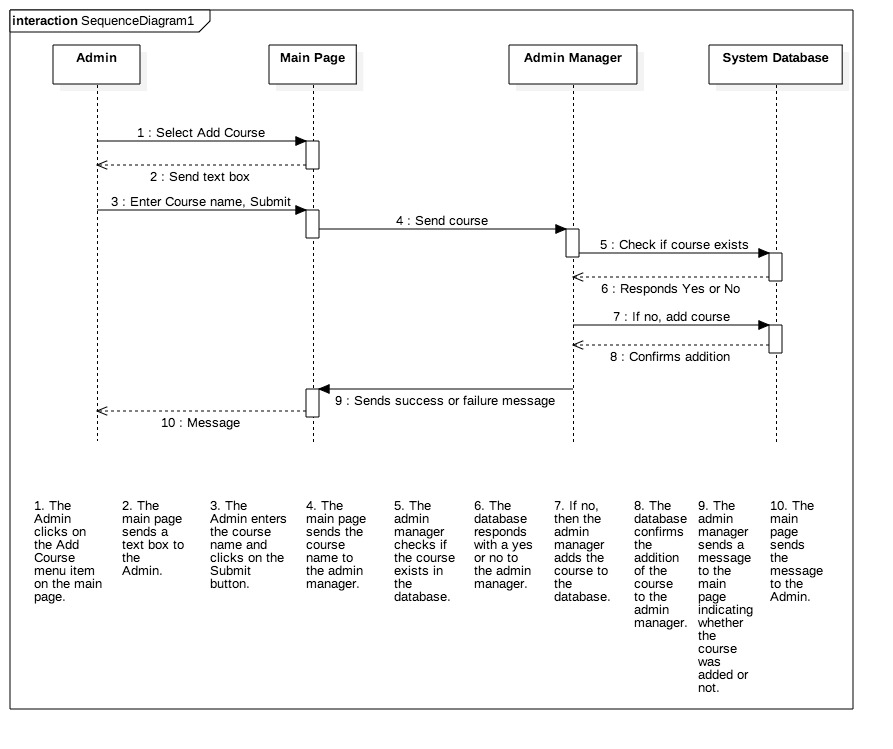
**13. Delete Account**



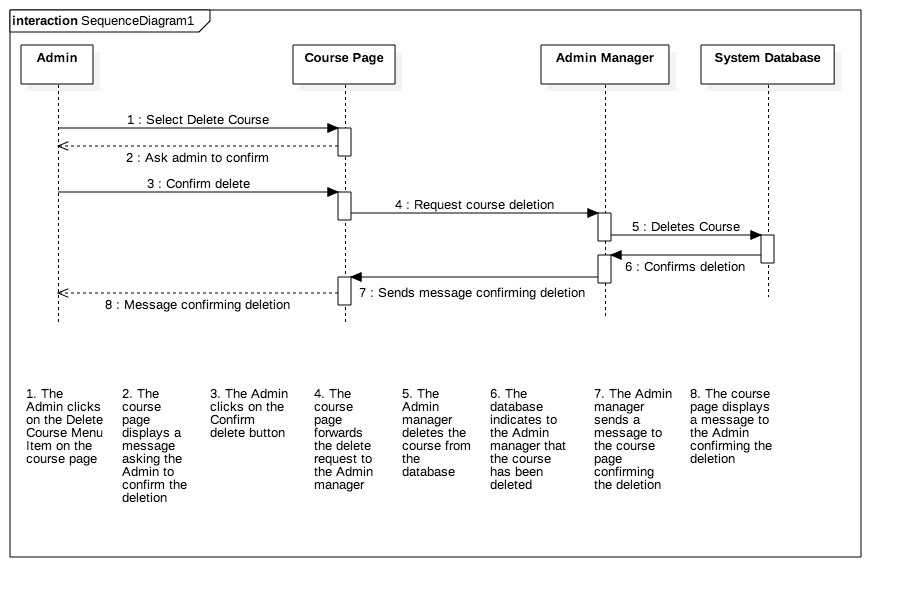
**14. Delete Member Account**



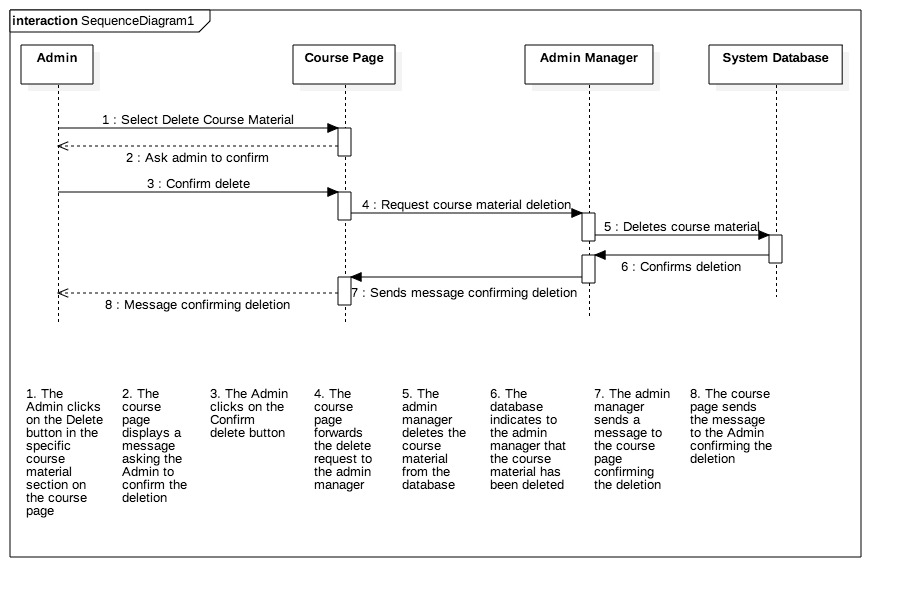
**15. Add Course**



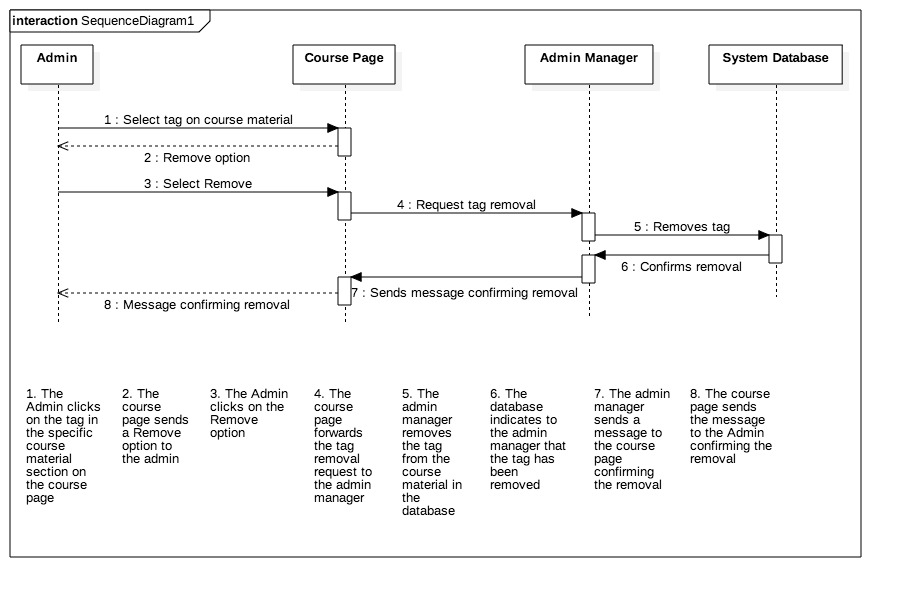
**16. Delete Course**



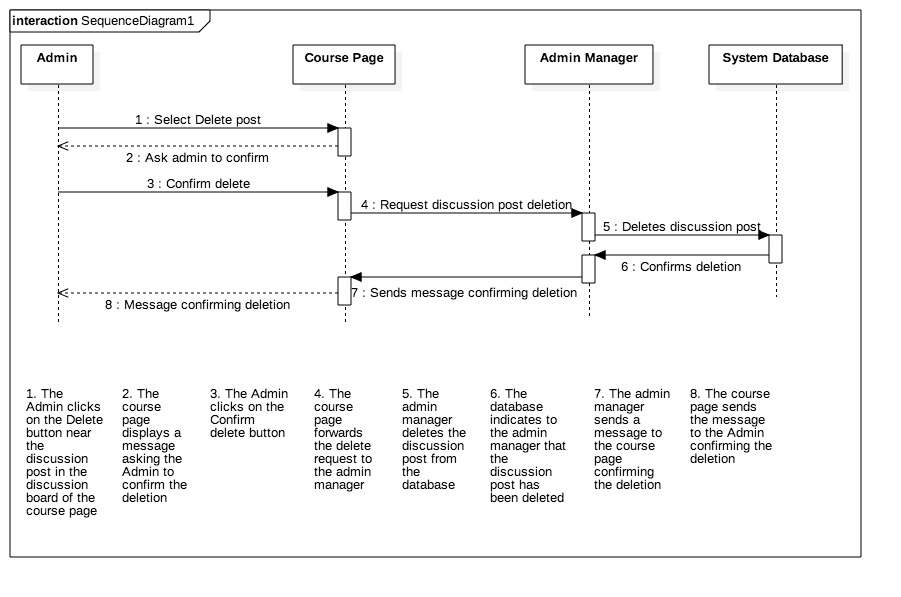
**17. Delete Course Material**



**18. Remove Tag**



**19. Delete Discussion Post**



**20. Logout**

