Software Requirement Specification

IU Committee

Team Banana

October 29, 2012

Table of Contents

List of Figures

1.0 Introduction

There are many committees across the Indiana University System. This project proposes to create a web-based committee management system that will allow committees to function in a more uniform manner, increase their productivity, and allow for greater transparency to the committee's work. This program will include features for scheduling meetings, managing committee documents and members, and allow the public to easily find committee meeting members, and minutes. Since this application will be web-based, there is no special software committee members will need to use the application. Also, since the users are already a part of the IU system, they will not need a different logon and will use the same method to login on other IU systems (such as the Oncourse system).

2.0 Domain description

2.1 Glossary

agenda a list of topics or decisions to be discussed at a committee meeting.

chair the leader of a committee. This person is also a committee admin.

co-chair the leader of a committee if there are two or more who share the role. This person is also a committee admin.

committee a person or group of people that works to complete some goal, usually an administrative goal.

committee admin this person has the ability to add/modify/delete meetings, discussion items, and committee documents. This person is either a chair, a co-chair, or a convener.

convener the leader of a committee until the chair is decided. His/her main job is to schedule and preside over the committee's first meeting. This person is also a committee admin.

division the area of the university hierarchy. Can be either the university itself or a campus, school, or unit. Each division except the university has a parent division.

document any computer information file, (e.g. word document, text file, music, jpeg images, pdf, etc).

ex officio a non-voting member of a committee. He is a member of the committee by reason of some office or position he holds.

IT admin the person responsible for adding users to the system and managing divisions.

meeting normally a gathering of committee members. This is where much of a committee's business happens. Meetings can be virtual, without an actual physical meeting.

minutes a record of what happened at a committee meeting.

non-voting member a person who belongs to a committee that does not have the power to vote.

super committee admin the person who is responsible for to add/remove/modify committees and assign users as members of a committee.

voting member a person who belongs to a committee and has the power to vote.

3.0 Function description

3.1 Access levels

Public:

The public documents can be accessed by any user.

Protected:

The protected documents can be accessed only by member who current and future members of the committee they belong to. (E.g. members can access protected documents only if they were created before or during when they sat on the committee.)

Private:

The private documents can be accessed only by users who are/were sat on the committee when the document was created (e.g. members can access private documents only if the document was created while they sat on the committee).

3.2 Actor Profiles (User roles)

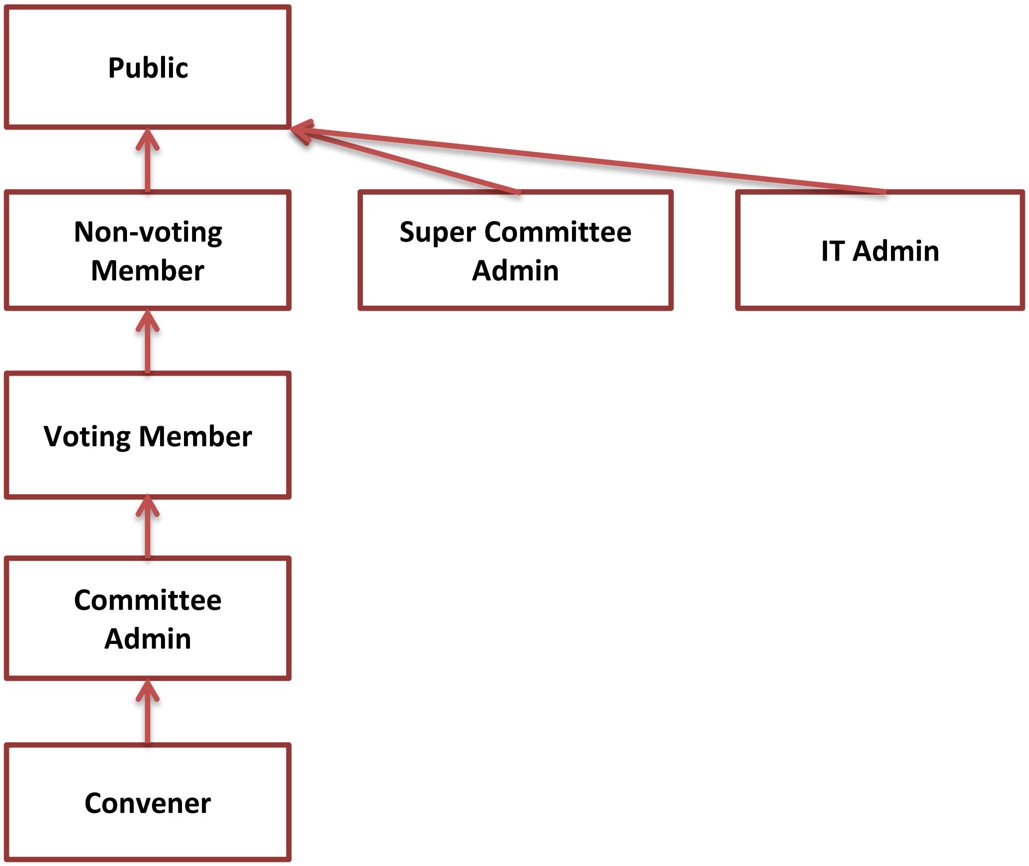

Figure 1 Actor Heirarchy

Public:

The public user can only view basic committee information such as the committee's current members, committee's charge and constitution, and any public document.

Non-Voting Member:

The non-voting member user is a part of at least one committee. They can do everything the public user can, and they can also read and comment on discussion items of the committee they belong to.

Voting Member:

The voting member user is a part of at least one committee. They can do everything the public and non-voting member can, and they can also vote on discussion items of the committee they belong to.

Committee Administrator:

The committee administrator user is a part of at least one committee. They can do everything the public, non-voting member and voting member can. They can also add, remove, and modify a meeting, a document, or a discussion item of the committee they belong to.

Convener:

The convener user is a part of at least one committee. They can do everything the public, non-voting member, voting member and committee administrator. They have the ability to assign the committee administrator(s) of the committee they belong to. Once the convener assigns the committee administrator(s) of the committee then the convener becomes a voting member.

Super Committee Administrator:

The super committee administrator user can add, remove, and modify committees and the members of the committees. They can also assign the convener of committees.

IT Administrator:

The IT administrator user can add divisions and users to the system. They can also assign the super committee administrator role to users.

3.3 Use Case Diagram

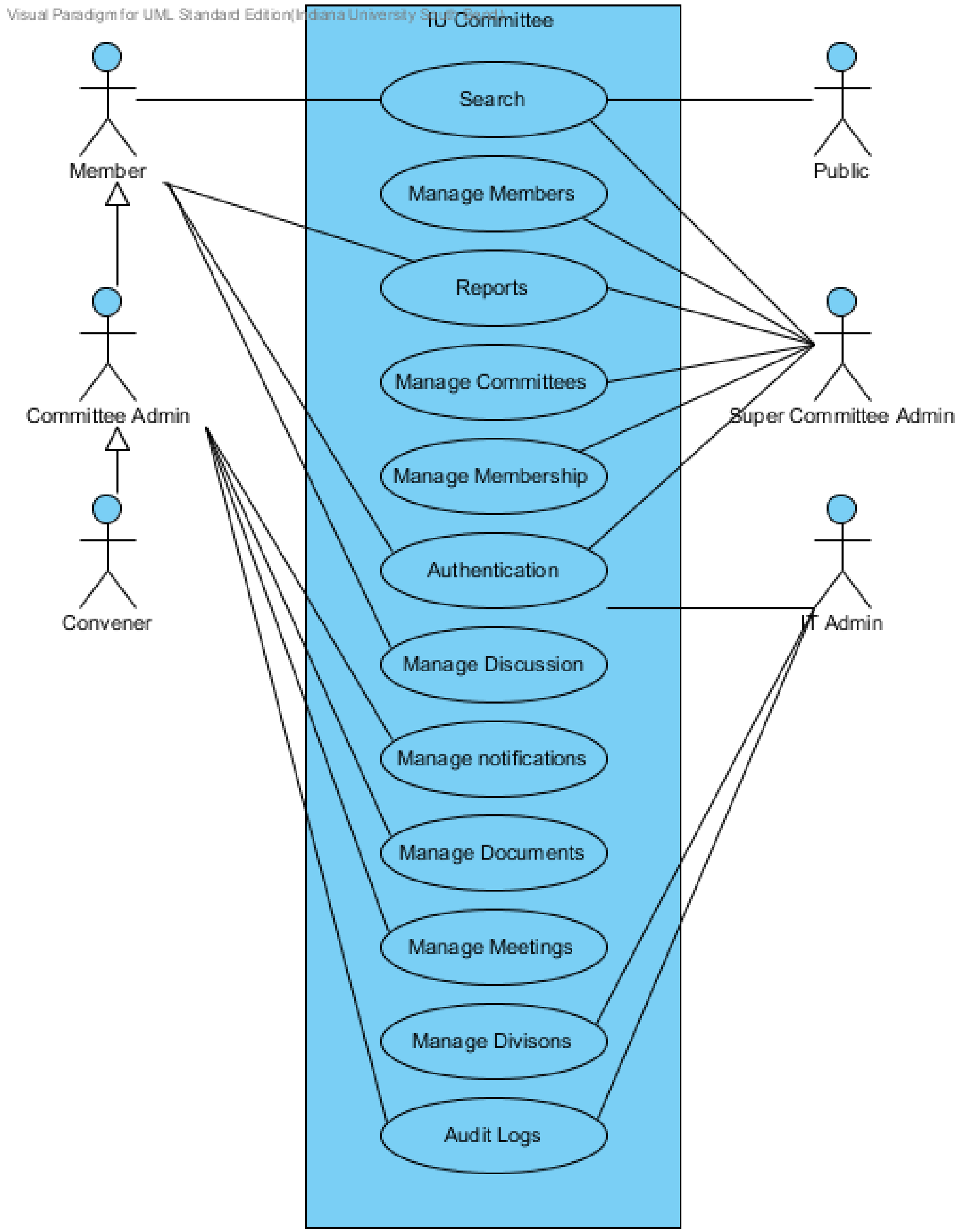

Figure 2 Use case diagram

3.4 Use Case Descriptions

3.4.1 Manage Documents

Actors: Super Committee Admin

Brief Description

The Manage Documents use case allows CAs to add, modify, and delete documents for committees of which they are the CA.

Step-by-Step Description

1. The CA is logged into the system.
2. Determine the committees of which the CA is a committee administrator and display list to CA
3. The CA selects a committee to which he has access.
4. Find all documents that are marked public that belong to the selected committee.
5. Find all documents that are marked protected that belong to the selected committee.
6. Find all documents that are marked privated and belong to the selected committee and were created while the CA was an administrator of the selected committee.
7. Display a list of the documents from steps 4, 5, and 6.
8. Modify or delete document
   a. The CA selects a document to modify or delete
   b. The selected documents attributes are retreived from the database and presented to the CA.
   c. Document attributes include name, access level, tags, and category.
   d. If access level is public include category.
   e. The CA modifies the attributes and selects save or selects delete document.
   f. Update the document attributes in the database or delete the document from the database based on the CA's selection.
9. Add new document
   a. The CA selects make new document.
   b. Prompt the CA to upload a document and to enter the document attributes.
   c. The CA uploads document and enters document attributes.
   d. Create a new document and store it to the database.

Figure 3 Description of Manage Documents use case.

Scenario 1: Modify document

The CA logs in into the system.
1. Determine the committees of which the CA is a committee administrator and display list to CA
The CA selects a committee to which he has access.
2. Find all documents that are marked public that belong to the selected committee.
3. Find all documents that are marked protected that belong to the selected committee.
4. Find all documents that are marked privated and belong to the selected committee and were created while the CA was an administrator of the selected committee.
5. Display a list of the documents from steps 4, 5, and 6.
The CA selects a document to modify or delete
6. The selected documents attributes are retreived from the database and presented to the CA.
Document attributes include name, access level, tags, and category.
The CA modifies the attributes and selects save document.

7. The updated attributes are stored in the database.

Scenario 2: Delete document

The CA logs in into the system.
1. Determine the committees of which the CA is a committee administrator and display list to CA
The CA selects a committee to which he has access.
2. Find all documents that are marked public that belong to the selected committee.
3. Find all documents that are marked protected that belong to the selected committee.
4. Find all documents that are marked privated and belong to the selected committee and were created while the CA was an administrator of the selected committee.
5. Display a list of the documents from steps 4, 5, and 6.
The CA selects a document.
6. The selected documents attributes are retreived from the database and presented to the CA.
7. Document attributes include name, access level, tags, and category.
The CA selects delete document.
8. Delete Document from the database.

Scenario 3: Add new document

The CA logs in into the system.
1. Determine the committees of which the CA is a committee administrator and display list to CA
The CA selects a committee to which he has access.
2. Find all documents that are marked public that belong to the selected committee.
3. Find all documents that are marked protected that belong to the selected committee.
4. Find all documents that are marked privated and belong to the selected committee and were created while the CA was an administrator of the selected committee.
5. Display a list of the documents from steps 4, 5, and 6.
The CA selects make new document.
6. Prompt the CA to upload a document and to enter the document attributes.
The CA uploads document and enters document attributes.
7. Create a new document and store it to the database.

3.4.2 Reports

Actor: Super Committee Admins (SCA), Committee Admin (CA), Member

Brief Description

The Reports use case enables a SCA or CA or Member to see listing of certain committees, users, or discussions.

Step-by-Step Description

1. Determine reports available to the current role.
2. Display list of reports.
3. The user selects a report.
4. List of committees without enough members
    a. Determine which committee belong the SCA's division(s)
    b. For each committee found count the members and compare to min requirement
    c. Display list of committee to SCA
5. List of previous chairs for all committees
    a. Determine which committee belong to the SCA's division(s)
    b. For each committee find all users who had the chair role at any time.
    c. Display list of users and committees to SCA
6. List of all discussion items without votes but need votes
    a. Determine all of the discussion items that the CA has access to and have no votes, but need votes.
    b. Display list of discussion items to CA
7. List of all discussion items that were approved
    a. Determine all of the discussion items that the CA or Member has access to and were approved.

Figure 4 Description of Report use case

Scenario 1: Display "Committee without enough members report"

1. An SCA wants to see a list of a committee without enough members.
2. The SCA selects "List of committee without enough members"

3.4.3 Manage Meetings

Actors: Committee Admin (CA), Convener

Brief Description

The Manage Meetings use case allows CA and Convener to add, remove, or modify a meeting and to add a final agenda.

Step by Step

The CA/Convener user is already logged into the system.

1. Determine the committees of which the CA/Convener is a CA/convener and display a list to the CA/Convener.
2. The CA/Convener selects a committee to which he has access.
3. Find all meetings for the selected committee.
4. Add a new meeting.
    a. The CA/Convener selects add a new meeting.
    b. Prompt the CA/Convener to input the date, time, and location of the meeting.
    c. The CA/Convener inputs the date, time, and location of the meeting.
    d. Create a new meeting and store it to the database.
5. Remove or modify a meeting or add a final agenda.
    a. The CA/Convener selects a meeting to remove or modify.
    b. The selected meeting's date, time, location and final agenda is retrieved from the database and presented to the CA/Convener.
    c. The CA/Convener modifies the meeting's date, time, location, and/or final agenda and selects save or selects delete meeting.
    d. If the CA/Convener selects delete meeting and the meeting has no discussions associated with it then it is deleted from the system else the CA/Convener is prompted that there is still at least one discussion associated with that meeting.
    e. If the CA/Convener selects save then update in the database the meeting's new date, time, location, and final agenda.

Figure 5 Description of Manage Meetings use case

Scenario 1: The CA adds a new meeting

The CA user is already logged into the system.

1. Determine the committees of which the CA is a CA and display the list to the CA.
2. The CA selects a committee to which he has access.
3. Find all meetings for the selected committee.
4. The CA selects add a new meeting.
5. Prompt the CA to input the date, time, and location of the meeting.
6. The CA inputs the date, time, and location of the meeting.
7. Create a new meeting and store it to the database.

Scenario 2: Convener modifies a meeting

The Convener user is already logged into the system.

1. Determine the committees of which the Convener is a convener and display a list to the Convener.
2. The Convener selects a committee to which he has access.

3. Find all meetings for the selected committee.
4. The Convener selects a meeting to modify.
5. The selected meeting's date, time, location and final agenda is retrieved from the database and presented to the Convener.
6. The Convener modifies the meeting's date, time, location, and/or final agenda and selects save.
7. Update in the database the meeting's new date, time, location, and final agenda.

Other Scenarios:

- Remove a meeting.
- Add a final agenda (just modify a meeting).
- Exception Scenarios
- Remove a meeting when CA wanted to just modify a meeting.

3.4.4 Manage Discussions

Actors: Committee Admin (CA), Convener, Voting Member, Non-Voting Member

Brief Description

The Manage Discussions use case allows CA and Convener to add, remove or modify a discussion. It also allows committee voting and non-voting members to read and comment on discussions and voting members to vote on discussions

Step by Step

The Member, CA, or Convener user is already logged into the system.

1. Determine the committees of which the Member/CA/Convener is a member/CA/convener and display a list to the Member/CA/Convener.
2. The Member/CA/Convener selects a committee to which he has access.
3. Find all discussions for the selected committee.
4. Add new discussion if the user is CA/Convener
   a. The CA/Convener selects for which meeting the discussion item is for.
   b. The CA/Convener is prompted to input the title and description of the discussion item.
   c. The CA/Convener inputs the title and description of the discussion item.
   d. The CA/Convener selects if the discussion item is to be voted on or not.
   e. If the discussion item is to be voted on then the CA/Convener inputs the date that voting ends.
   f. The CA/Convener selects the documents, if any, to be added to the discussion item.
   g. Create a new discussion and store it to the database.
   h. A notification is created for the committee members that a new discussion is added.
5. Remove or modify a discussion if the user is CA/Convener
   a. The CA/Convener selects the discussion item they want to edit.
   b. The CA/Convener is presented with the discussion's title, description, if it is able to be voted on and the voting end date, and list of documents.
   c. The CA/Convener modifies the discussion's title, description, if it is able to be voted on and the voting end date and list of documents and selects save or delete discussion.
   d. Update the discussion information in the database or delete the discussion from the database based on the CA/Convener's selection.
6. Read, comment, or vote on discussion if the user is a non-voting or voting member.
   a. The member selects a discussion they want to read, comment, or vote on.
   b. The member reads the discussion and clicks a "mark discussion read button."
      i. The database is updated to show that that member has read the discussion.
   c. If the user is a voting member they select the the "For", "Against", or "Maybe."
      i. The database is updated to show how the voting member voted.
   d. The member is presented with a text field to input a comment.
   e. The member adds text to the comment text field and submits the comment.
   f. The comment is added to the discussion and the database is updated.

Figure 6 Description of Manage Discussions use case

Scenario 1: CA add a new discussion

The CA user is already logged into the system.

1. Determine the committees of which CA is a CA and display a list to the CA.
2. The CA selects a committee to which he has access.

3. Find all discussions for the selected committee.
4. The CA selects for which meeting the discussion item is for.
5. The CA is prompted to input the title and description of the discussion item.
6. The CA inputs the title and description of the discussion item.
7. The CA selects if the discussion item is to be voted on or not.
8. If the discussion item is to be voted on then the CA inputs the date that voting ends.
9. The CA selects the documents, if any, to be added to the discussion item.
10. Create a new discussion and store it to the database.
11. A notification is created for the committee members that a new discussion is added.

Scenario 2: Member adds a comment

The Member user is already logged into the system.

1. Determine the committees of which the Member is a member and display a list to the Member.
2. The Member selects a committee to which he has access.
3. Find all discussions for the selected committee.
4. The member selects a discussion they want to comment on.
5. The member is presented with a text field to input a comment.
6. The member adds text to the comment text field and submits the comment.
7. The comment is added to the discussion and the database is updated.

Other Scenarios

- Modify a discussion.
- Remove a discussion.
- Read a discussion.
- Vote on a discussion.

Exception Scenarios

- Member voted wrong.
- Modify the wrong discussion.
- Remove a discussion instead of just modifying the discussion.

3.4.5 Search

Actors: Public, Member

Brief Description

The Search use case allows the Public users to search public documents. It also allows Members to search documents in committees that the member is a part of.

Step by Step

The Member is already logged into the system. All other users are considered Public.

1. Determine the committees the Member is a part of.
2. The Public/Member user is prompted to enter a keyword to search.
3. Gets the documents that the Public/Member user is able to see.
4. Check the tags and filenames of the retrieved documents with the keyword that the Public/Member user inputted.
5. If there is a match the file is presented to the Public/Member user as a link to download.

Figure 7 Description of Search use case

Scenario 1: Search

The Member logs into the system.

1. Determine the committees the Member is a part of.
2. Prompt the user to enter keyword to search.
3. Gets the documents that the Member user is able to see.
4. Check the tags and filenames of the retrieved documents with the keyword that the Member user input.
5. If there is a match present the file to the member as a link to be downloaded.

3.4.6 Authentication

Actors: IT Admin, CSA, Convener, Committee Admin, Members

Brief Description:

Users who are to be active members of the IU Committee service are authenticated using their IU Account. When logging in the user will be redirect to the IU CAS service, then upon successful login be redirected to the IU Committee software. When logging out the system will destroy the user session to prevent any further access to protected pages and content.

Step by Step

1. The user visits the IU Committee system.
    a. The system initially recognizes them initially as a public role.
    b. The User presses the “Login” button.
    c. The user is transferred to the CAS.
        i. The User enters IU ID and Password into the text fields
        ii. CAS system verifies this is correct.
        iii. Upon successful authentication in CAS the user is redirected to IU Committee and IU CAS passes token along with them.
        iv. IU Committee system processes token and confirms authentication with CAS.
        v. The authentication module confirms the user and role and redirects to their user dashboard.
2. The user makes any page request.
    a. The authentication module confirms the existing session.
        i. If the user session is invalid the session is destroyed and user is redirected to the CAS login and follows the flow in step 1.3.
        ii. If the user session is valid the authentication module allows the page request to be processed.
3. The user is logged in and can see the “Logout” button.
    a. The user presses the “Logout” button.
    b. Destroy the existing user session.
    c. Redirect the user to the public home page of IU Committee

Figure 8 Description of Authentication use case

Scenario 1: Login

1. A user clicks on the login button shown on any page of the system to a non-authenticated user.
2. Set appropriate settings and redirects the user to the IU CAS system.
3. The user authenticates with their IU Credentials on the IU CAS.
4. IU CAS Redirects the user back to the IU Committee software.
5. The IU Committee system processes necessary tokens and confirms the authentication is legitimate with IU CAS.
6. Find the associated user in the database and assigns the session.
    a. If an associated user is not found, it creates a new one and saves the data.
7. Update the last login information.
8. Redirect the user to the main dashboard.

Scenario 2: Logout

1. The user presses the logout button.
2. Destroy the current session for the user.
3. Redirect the user to the public homepage.

3.4.7 Manage Divisions

Actors: IT Admin

Brief Description:

The IT admin user needs to be able to manage the different levels of committees. Including University, Campus, College, Department/Unit. He can add, delete and edit divisions.

Step by Step:

1. The IT admin is already logged in.
2. IT admin clicks "Manage Divisions".
3. They are presented with a searchable list of current divisions organized by hierarchy.
4. The IT Admin clicks "Add a Division"
   a. The IT admin enters the name of the new division.
   b. The IT admin selects what the University the new division belongs to.
   c. The IT admin selects what Campus the new division belongs to.
   d. The IT admin selects what College (if applicable) the new division belongs to.
   e. The IT admin selects what Department/Unit (if applicable) the new division belongs to.
   f. The IT admin is taken to a confirmation screen with all the relevant info.
      i. If the info is correct they click "confirm" and proceed to step 7 of the flow.
      ii. If the info is incorrect they are redirected to the "edit" for the division found in step 6.
   g. The IT admin is returned to the main listing of committees.
5. The IT Admin selects a "Delete" button on a corresponding committee.
   a. To confirm a "deletion" by typing the name of the division the wish to delete.
   b. The division is "flagged" as deleted by the system. Files and settings still exist but become in accessible.
   c. All committees within the division are flagged as "archived".
   d. The admin is returned to the main listing of committees.
6. The IT Admin selects a "Modify" button on a corresponding committee.
   a. The IT admin edits the name of the new division.
   b. The IT admin edits what the University the new division belongs to.
   c. The IT admin edits what Campus the new division belongs to.
   d. The IT admin edits what College (if applicable) the new division belongs to.
   e. The IT admin edits what Department/Unit (if applicable) the new division belongs to.
   f. The IT admin is taken to a confirmation screen with all the relevant info.
      i. If the info is correct they click confirm and proceed to step 7.
      ii. If the info is incorrect they are taken to the beginning of the edit flow.
   g. The IT admin is returned to the main listing of committees.
7. Return the IT Admin to the main dashboard when done with adding new divisions.

Figure 9 Description of Manage Divisions use case

Scenario 1: Add Division (Department/Unit)

The IT Admin must be logged in and authenticated.

1. The ITA clicks "Add New Division" button.
2. Direct the ITA to the GUI to add a new Division.
3. The ITA can select the Parent University from a dropdown, or adds a new University.
4. Update GUI and add all the children campuses of the selected university to a dropdown.
5. The ITA selects a campus or adds a new campus.
6. Update the GUI and add all the children colleges of the selected campus to a dropdown.
7. The ITA selects a college or adds a new college.
8. The ITA adds a new department or unit.
9. The parent division's id and all other entered info are sent to the system.
10. Validate the information.
11. Save the new division.
12. Redirect the ITA a fresh "Add New Division" screen.

Scenario 2: Modify Division

The IT Admin must be logged in and authenticated.

1. The ITA searches or navigates through the University, Colleges, Departments, and Units.
2. The ITA finds the Division they wish to edit.
3. The ITA clicks a corresponding edit button on the division the wish to edit.
4. The ITA can modify the division's information, such as parent division and name.
5. The ITA clicks the "Save" button.
6. The information is sent to the server.
7. Validate the information.
8. Save the updated information.
9. The ITA is returned to the main divisions GUI.

Scenario 3: Delete Division

The IT Admin must be logged in and authenticated.

1. The ITA searches or navigates through the University, Colleges, Departments, and Units.
2. The ITA finds the Division they wish to delete.
3. The ITA clicks a corresponding delete button on the division the wish to delete.
4. The system presents a confirmation box to the ITA to make sure the deletion is correct.
5. If the user clicks cancel close the confirmation and do not delete.
6. If the deletion is confirmed, the division is removed from the database along with all related elements.
7. The ITA is returned to the main divisions GUI.

3.4.8 Audit logs

Actors: IT Administrator and Committee Administrator

Brief Description

The view audit logs use case allows the IT Administrator to view auditing information that is kept for the system. It also allows a Committee Administrator (CA) of a committee to view the auditing information kept for various tasks that are done within his/her committee.

Step by step

1. The IT Administrator or Committee Administrator logs into the IU Committee system.
2. If the user is a CA:
    a. Determine the committees of which the CA is a committee administrator.
    b. Display the list of committee(s) to the CA.
    c. The CA selects a committee to which he/she has access to.
    d. The CA clicks “View Audit Logs”.
    e. The CA is presented with the options to display information about:
        i. The votes on discussion items.
        ii. If a user has read a discussion item.
3. If the user is the IT Administrator:
    a. The IT Administrator clicks “View Audit Logs”.
    b. The IT Administrator is presented with the options to display information about:
        i. System errors.
        ii. System usage.
4. When the user is done choosing the information he/she wishes to display, the user clicks “Generate Log Report.
5. Fetch the audit log file from the database.
6. Create an audit log report.
7. Display audit log report file in the web browser.

Figure 10 Description of View Audit Logs use case

Scenario 1: IT Admin view

1. The IT Admin must be logged in and authenticated.
2. The IT Administrator clicks the “View Audit Report” button from their interface.
3. The IT Administrator is taken to a new page that displays a checkbox with the various types of information they can display.
4. The IT Administrator checks the “View System Errors” field.
5. The IT Administrator submits this form.
6. Fetch the auditing log file from the database.
7. Create and open an auditing log report file.
8. Read through each entry in the auditing log file.
9. Write all the entries relevant to “System Errors” read from the auditing log file to the auditing log report file.
10. Display the audit log report file in the browser.

Scenario 2: Committee Admin View

1. The Committee Administrator must be logged in, authenticated and already selected the committee in which he/she has access to.
2. The Committee Administrator clicks the "View Audit Report" button from their interface.
3. The Committee Administrator is taken to a new page that displays a checkbox with the various types of information they can display.
4. The Committee Administrator checks the "View Votes on Discussion Items" field.
5. The Committee Administrator submits this form.
6. Fetch the auditing log file from the database.
7. Create and open an auditing log report file.
8. Read through each entry in the auditing log file.
9. Write all the entries relevant to "View Votes on Discussion Items" read from the auditing log file to the auditing log report file.
10. Display the audit log report file in the browser.

3.4.9 Manage Notifications

Actors: Committee Administrator

Brief Description

The manage notifications use case allows the system to send a notification to the committee members that belong to the committee that the notification is for.

Step by Step

1. The committee administrator creates a new discussion item and a notification is generated.
2. The system reads this notification from the database.
   a. Get the discussion item ID.
   b. Get the meeting ID that is associated with the discussion item ID.
   c. Get the committee ID associated with the meeting ID.
   d. Get all the users that are members of the committee associated with committee ID.
3. Send this notification to every member of the committee that the discussion item was created for.

Figure 11 Description of Manage Notifications use case

Scenario 1: Create discussion item

The Committee Administrator must be logged in, authenticated and already selected the committee in which he/she has access to.

4. The Committee Administrator creates a new discussion item.
5. A new notification is generated based on the information from the discussion item.
6. Go to this notification item in the database.
   a. Get the discussion ID of the discussion item.
   b. Get the meeting ID that is associated with the discussion ID.
   c. Get the committee ID that is associated with the meeting ID
   d. Get all of the users that are members of the committee that is associated with the committee ID.
7. The system sends this notification to all the users that belong to the committee that the discussion item was created for.

3.4.10 Manage Committes

Actors: Super Committee Admin (SCA)

Brief Description

The manage committee use case allows the SCA to create, edit, remove, and archive committees the SAC divisions.

Step by Step

1. The SCA creates a new committee.
   a. The SCA chooses campus from the list.
   b. The SCA adds a new committee by clicking add new button.
   c. The SCA then types the name for the committee.
   d. The SCA clicks save after filling all required information.
   e. The SCA verifies by clicking confirm button in the confirmation notice.
   f. After the confirmation of SCA, create a record for the committee.
   g. The system records this action in Audit file.
   h. The SCA receives the successful notice from the system.
2. The SCA edits a committee.
   a. The SCA chooses campus from the list.
   b. The SCA chooses a committee from the list.
   c. The SCA then changes the information of the committee.
      i. The SCA can modify the committee's charge.
      ii. The SCA can modify the committee's constitution.
      iii. The SCA can modify the Convener or CA of the committee.
      iv. The SCA can modify the name of the committee.
   d. The SCA hits save button after filling all required information.
   e. The SCA verifies by clicking confirm button in the confirmation notice.
   f. After the confirmation of SCA, the system applies the change on the record of the committee.
   g. The system records this action in Audit file.
   h. The SCA receives the successful notice from the system.
3. The SCA removes a committee.
   a. The SCA chooses division from the list.
   b. The SCA chooses a committee from the list.
   c. The SCA validates the restriction by clicking validate button.
   d. The SCA clicks remove button after validating successfully.
   e. The SCA verifies by clicking confirm button in the confirmation notice.
   f. After the confirmation of SCA, removes the record for the committee.
   g. The system records this action in Audit file.
   h. The SCA receives the successful notice from the system.

Figure 12 Description of Manage Committees use case

Scenario 1: Add new committee

1. The SCA click add new button.
2. The SCA chooses either campus/department/school from the drop down list.
3. The SCA then types name for the committee.
4. The SCA hits save button after filling all required information.

5. The SCA verifies by clicking confirm button in the confirmation notice.
6. After the confirmation of SCA, creates a record for the committee.
7. The system records this action in Audit file.
8. The SCA receives the successful created notice from the system.

Scenarios Alternate:

- The SCA might choose the wrong campus, department, school, or committee.
- The SCA might click save before type in the name of new committee.
- The validation might be unsuccessful. Then the SCA cannot remove a committee.
- The SCA might type in another data type which is different from text for committee name, charge, or constitution field.
- The SCA might click remove before validating the restriction.
- The SCA might cancel the action.

3.4.11 Manage Membership

Actors: Super Committee Admin (SCA)

Brief Description:

The Manage Membership use case, allows an SCA to add members to committees and manage committee membership.

Step by Step

1. The SCA adds member to the committee.
    a. The SCA chooses either campus/department/school from the list.
    b. The SCA adds new member for committee, which the SCA has selected, by clicking add button.
    c. The SCA then type the email address of the member.
        i. 1 The SCA also fills in the start date and end date.
        ii. 2 The SCA then assigns the role for the member.
    d. The SCA clicks save after filling in all required information.
    e. The system sends confirmation notice to the SCA
    f. After the confirmation, the system creates a record for the member.
    g. The system records this action in the audit logs.
    h. The system indicates successful notice to the SCA.
2. The SCA modifies membership of a member a committee.
    a. The SCA chooses either campus/department/school from the list.
    b. The SCA chooses a committee from the list.
    c. The SCA then changes the information of the member.
        i. The SCA can modify the role of a member and the start date and end date of a membership.
    d. The SCA clicks save after filling all required information.
    e. The SCA verifies by clicking confirm button in the confirmation notice.
    f. After the confirmation of SCA, the system applies the change on the record of the committee.
    g. The system records this action in the audit logs.
    h. The system indicates successful notice to the SCA.

Figure 13 Description of Manage Membership use case

Scenario: Add new member.

1. The SCA click add new button.
2. The SCA chooses either campus/department/school from the drop down list.
3. The SCA then types the email address for that committee member.
4. The SCA hits save button after filling all required information.
   a. If the email address is in correct form, go to step 5
   b. If the email address is in incorrect form
      i. The system shows up an error notice.
      ii. The SCA clicks “Accept” button to start over the process.(go to step 3)
5. The SCA verifies by clicking confirm button in the confirmation notice.
6. After the confirmation of SCA, create a record for the new member of committee.
7. The system records this action in audit logs.
8. The system indicates successful notice to the SCA

Scenarios Alternate:

- The SCA user might choose wrong either campus,department,school, or committee.
- The SCA user might assign wrong membership which is existing.
- The SCA user might click save before assign role for a member.
- The SCA user might click remove wrong member's role.
- The SCA user might cancel the action.

4.0 Data description

4.1 Entity class diagram

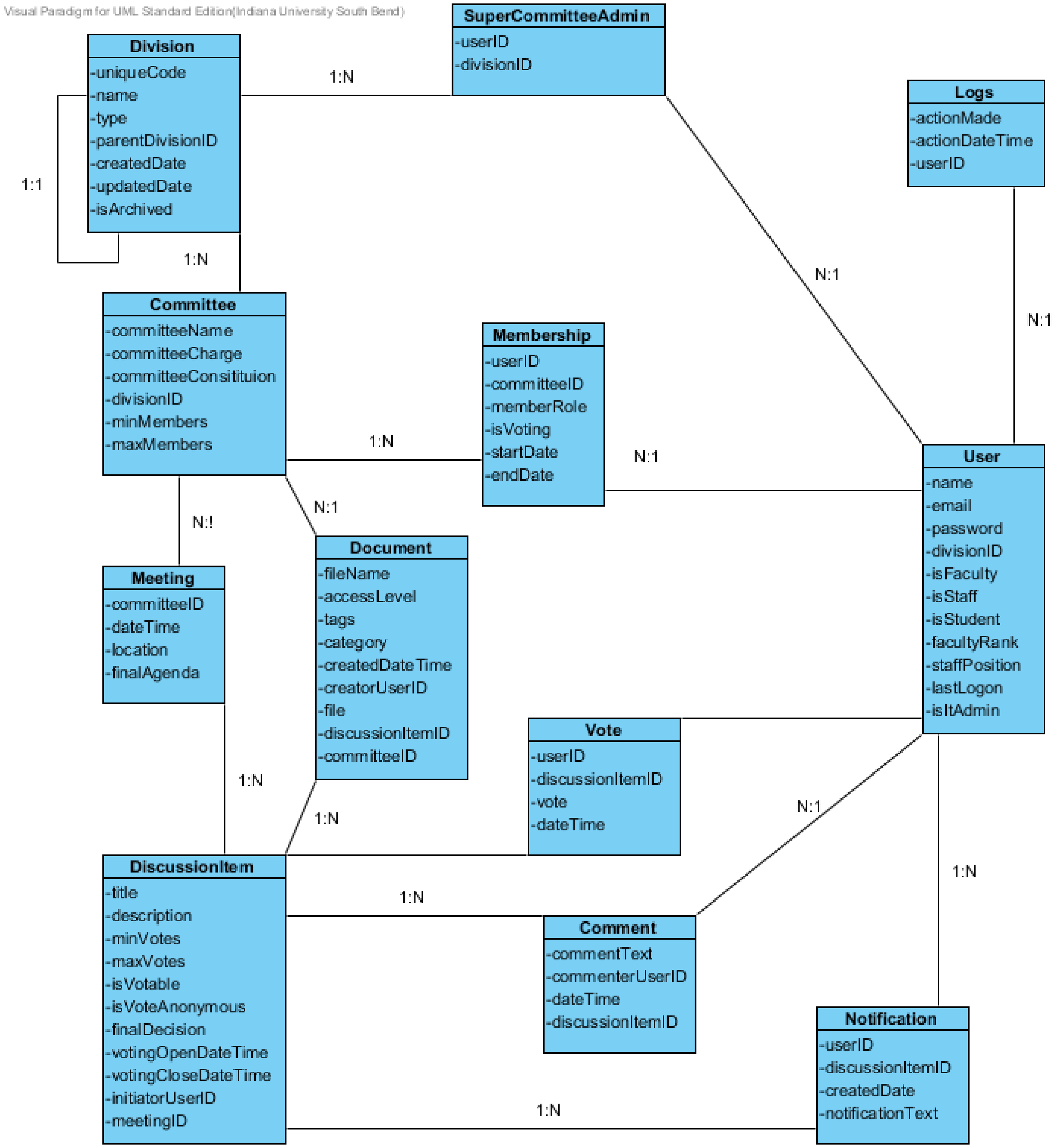

Figure 14 Entity Class Diagram

4.2 Entity class descriptions

4.2.1 Document

The documents are objects that can be added to any of the discussion items by the committee chair. Documents also stand alone in committees. Documents have an access level to control who can view them, tags for searching documents, and a category to group committee documents by.

4.2.2 Discussion Item

Discussions are an object for allowing members of the committee to comment, vote, and view on committee matters. Documents can be uploaded to the discussion.

4.2.3 Comments

Comments are a part of discussion items, and are created by any member of a committee.

4.2.4 Vote

Votes are a part of discussion items. This feature can be set as on or off by the creator of the discussion item, and only members who are set as voting members can create a vote.

4.2.5 Committee

The committee object holds the committee name, and division it belongs to, along with the charges, constitution, and minimun and maximum amount of members aloud to be in the committee at any time. This model is only created, updated, and deleted by Super Committee Admin levels.

4.2.6 Division

The division object is used for building the hierarchy of the systems Universities, Campuses, Colleges, and Departments/Units. It is only editable by the IT Admins. A committee can belong to any one of the divisions and its parents.

4.2.7 Super Committee Admin

This object models a special role that allows the defined user to manage committees for the defined division.

4.2.8 Logs

The object for modeling logs is used for audits and tracking changes and actions within the system.

4.2.9 User

The user object stores all of the necessary information about an individual user in the system. Including name, email, password, division they belong to and their position and status within the division. The last login is also stored.

4.2.10 Notification

The notifications object is a simple way to remind and track users and their actions within a committee. The related user, discussion items, and notification content are stored.

4.2.11 Membership

The membership object stores the user and the committee that user is a part of, along with their rights within that committee and the duration of their membership.

4.2.12 Meeting

The meeting object stores the committee, date and time, location and a final agenda for each meeting.

5.0 Software behavior description

The software will behave in two main way, either an authenticated member interface, or public interface. The public interface allows for any publically readable information to be viewed or searched. The authenticated member interface is much more complex, allowing role based access to features within the system.

A user is automatically assigned to the public role and interface upon visiting the system, they must successfully login and authenticate before being able to utilize the authenticated member interface.

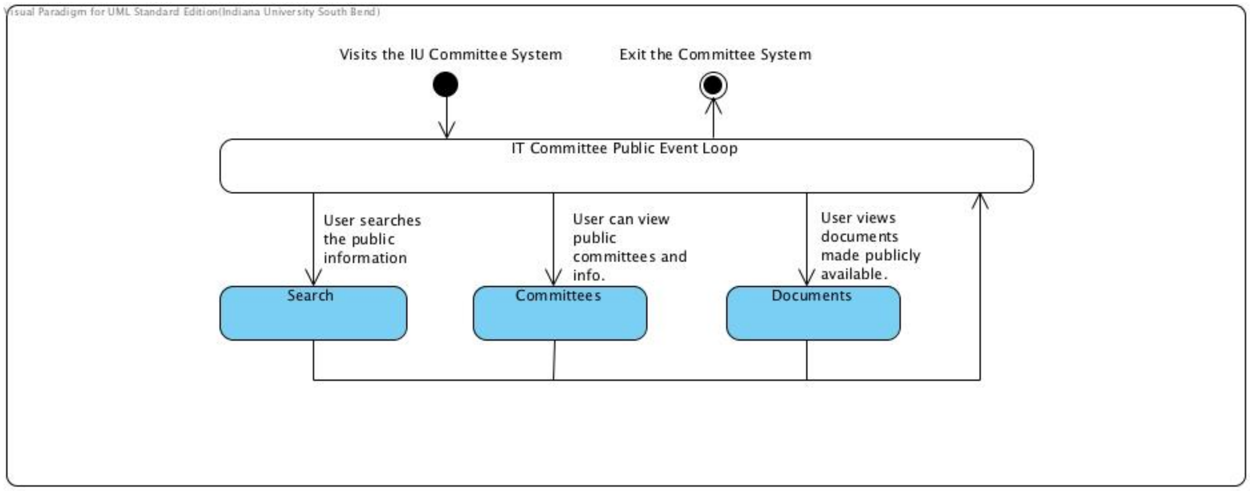
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
Visits the IU Committee System
Exit the Committee System
IT Committee Public Event Loop
User searches the public information
User can view public committees and info.
User views documents made publicly available.
Search
Committees
Documents

Figure 15 Public Role State Diagram

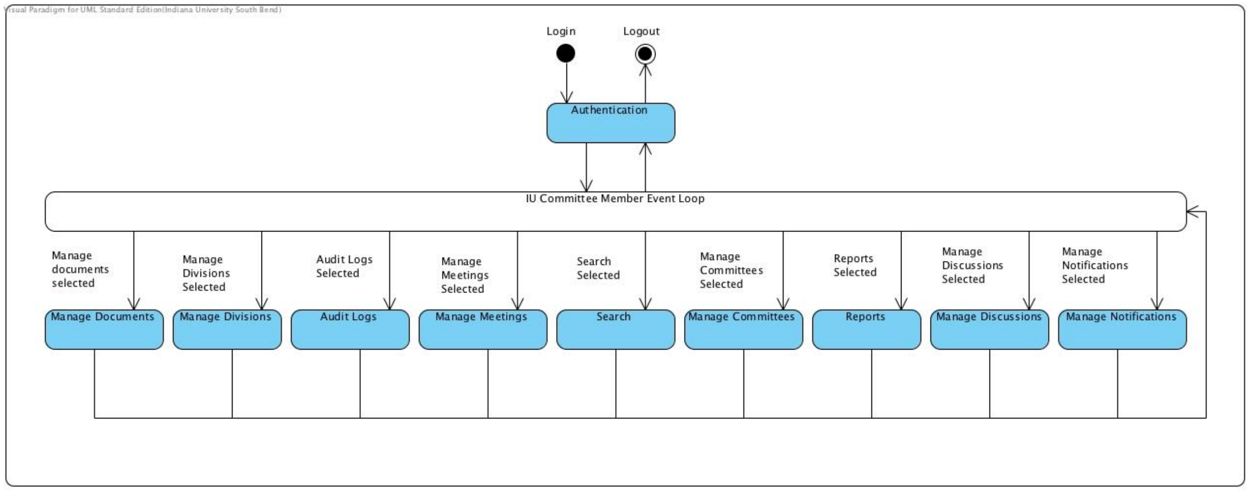
isual Paradigm for UML Standard Edition(Indiana University South Bend)
Login
Logout
Authentication
IU Committee Member Event Loop
Manage documents selected
Manage Divisions Selected
Audit Logs Selected
Manage Meetings Selected
Search Selected
Manage Committees Selected
Reports Selected
Manage Discussions Selected
Manage Notifications Selected
Manage Documents
Manage Divisions
Audit Logs
Manage Meetings
Search
Manage Committees
Reports
Manage Discussions
Manage Notifications

Figure 16 Authenticated Role State Diagram

6.0 Software interface description

6.1 External Machine Interfaces

The application will be built to run in any modern web browser, allowing for use across desktop, mobile and tablet interfaces.

6.2 External Systems Interfaces

The IU Committee Software will be utilizing the IU Central Authentication Service for integrating IU user accounts into the system. This provides an intuitive and known login mechanism for the users of the system, preventing need to create and manage a yet another set of user login information.

6.3 Human Interfaces

The human interface of the IU Committee system will be designed to be as user friendly as possible. The user will be provided with a menu to navigate through the system. Each user's menu will be based on the access level the user has. The public user role menu will have very minimal access to the system, while the super committee administrator and IT administrator will have a menu that has a great amount of control over the IU Committee system.

7.0 Complex calculation description

7.1 Control Classes

7.1.1 Documents control class

The documents control class provides access to committee documents. It ensures access control by determining the given users permissions. It serves as the gateway to the document entities. It uses the membership and committee entities for determining access control.

7.1.2 Reports control class

The reports control class will execute SQL queries against the underlying database to produce data for the requested reports. Access control will be performed via SQL query logic.

7.1.3 Meetings control class

The meetings control class provides access to committee meetings. It ensures access control by determining the given user permissions. It serves as the gateway to the meeting entities to be able to create, modify, and remove a meeting.

7.1.4 Discussions control class

The discussions control class provides access to committee discussions. It ensures access control by determining the given user permissions. It serves as the gateway to the meeting entities to be able to create, modify, and remove a discussion and to read, comment, and vote on.

7.1.5 Searches control class

The search control class provides access to search committee documents. It ensures access control by determining the given user permissions. It allows users to search the tags of documents using keywords.

7.1.6 Authentications control class

The authentication control does the complex interfacing between the IU Committee system and IU Central Authentication Service. It processes all the tokens and assigns a session to the user to their matching profile upon successful authentication. The authentication class also assures that the session is valid on every page request.

7.1.7 Divisons control class

The divisions control class enables the addition, modification, and deletion of divisions in the system. It also assures that only those of IT Admin level are able to make those additions and changes.

7.1.8 Committees control class

This class provides a method for managing committees such as adding, modifying committee information, and archiving committees in the system. Super Committee Admins can perform these tasks.

7.1.9 Memberships control class

This class allows Super Committee Admins to manage members of committees. The tasks performed by this class are adding members to committees, assigning a user to a committee, setting the start and end dates of the members term, and removing members.

7.1.10 Notifications control class

The notifications control class interacts with numerous classes to determine which members that the notification needs to be sent to. The notifications control class interacts with: the notifications entity class to get the discussion item ID, the discussion item entity class to get the meeting ID, the meeting entity class to get the committee ID, the membership class to determine the users that are associated with the committee ID, and the user class to send the notification to the members of the committee.

7.1.11 Logs control class

The logs control class is responsible for reading the data that is requested by the user from the database.

8.0 Use-case realization

8.1 Manage Documents

8.1.1 Scenario 1: Modify document

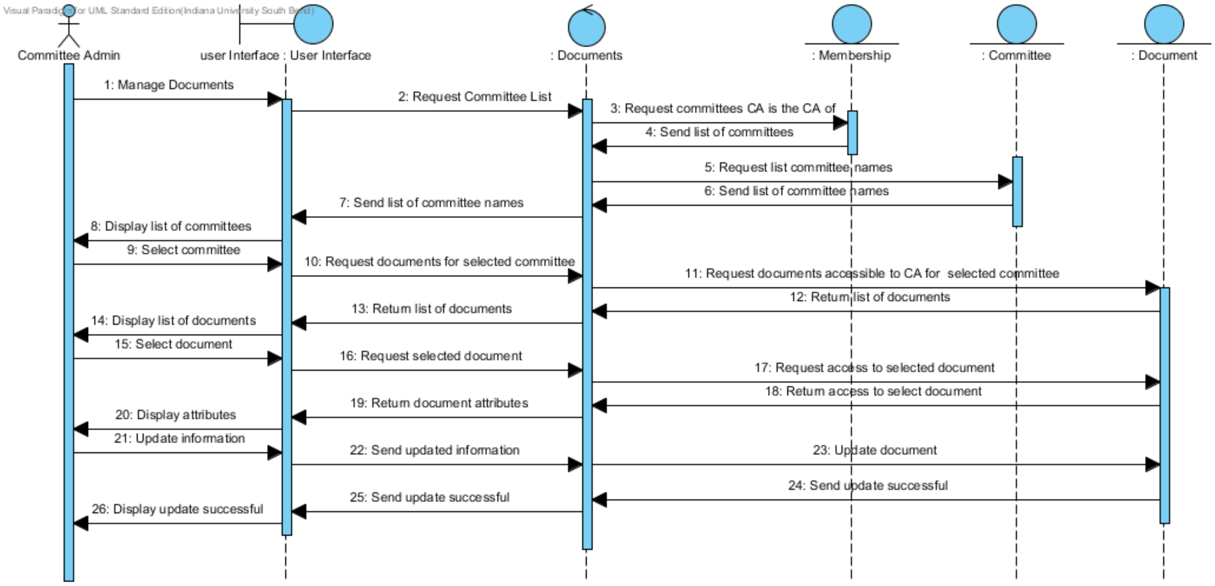
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
Committee Admin
user Interface : User Interface
: Documents
: Membership
: Committee
: Document
1: Manage Documents
2: Request Committee List
3: Request committees CA is the CA of
4: Send list of committees
5: Request list committee names
6: Send list of committee names
7: Send list of committee names
8: Display list of committees
9: Select committee
10: Request documents for selected committee
11: Request documents accessible to CA for selected committee
12: Return list of documents
13: Return list of documents
14: Display list of documents
15: Select document
16: Request selected document
17: Request access to selected document
18: Return access to select document
19: Return document attributes
20: Display attributes
21: Update information
22: Send updated information
23: Update document
24: Send update successful
25: Send update successful
26: Display update successful

Figure 17 Sequence diagram for Manage Documents scenario: modify documents

8.1.2 Scenario 2: Delete document

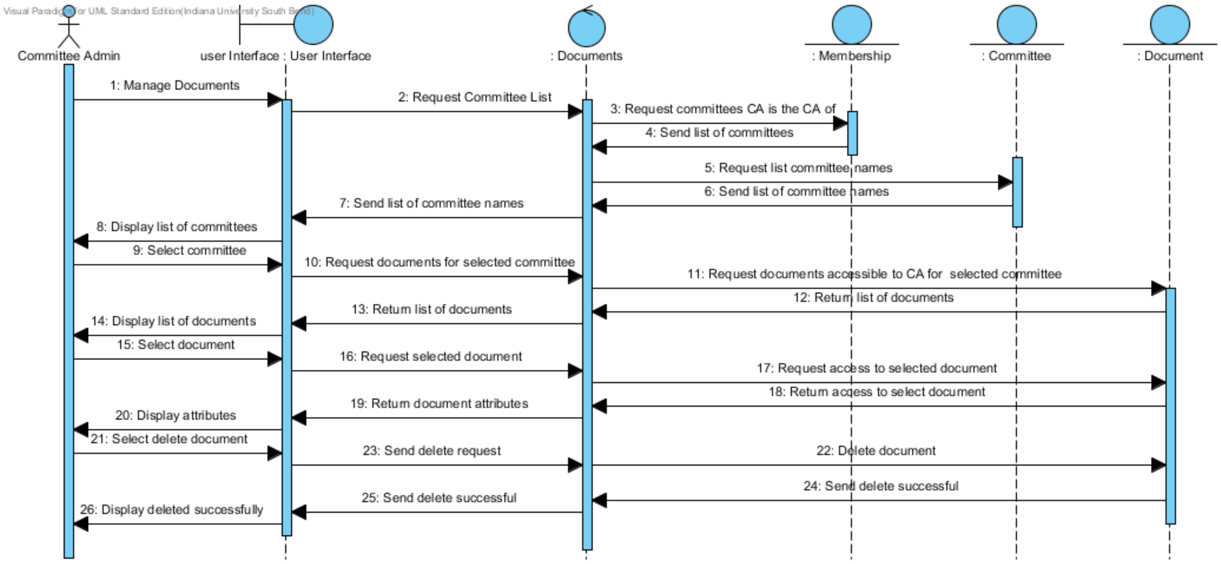
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
Committee Admin
user Interface : User Interface
: Documents
: Membership
: Committee
: Document
1: Manage Documents
2: Request Committee List
3: Request committees CA is the CA of
4: Send list of committees
5: Request list committee names
6: Send list of committee names
7: Send list of committee names
8: Display list of committees
9: Select committee
10: Request documents for selected committee
11: Request documents accessible to CA for selected committee
12: Return list of documents
13: Return list of documents
14: Display list of documents
15: Select document
16: Request selected document
17: Request access to selected document
18: Return access to select document
19: Return document attributes
20: Display attributes
21: Select delete document
22: Delete document
23: Send delete request
24: Send delete successful
25: Send delete successful
26: Display deleted successfully

Figure 18 Sequence diagram for Manage Documents scenario: delete document

8.1.3 Scenario 3: Add new document

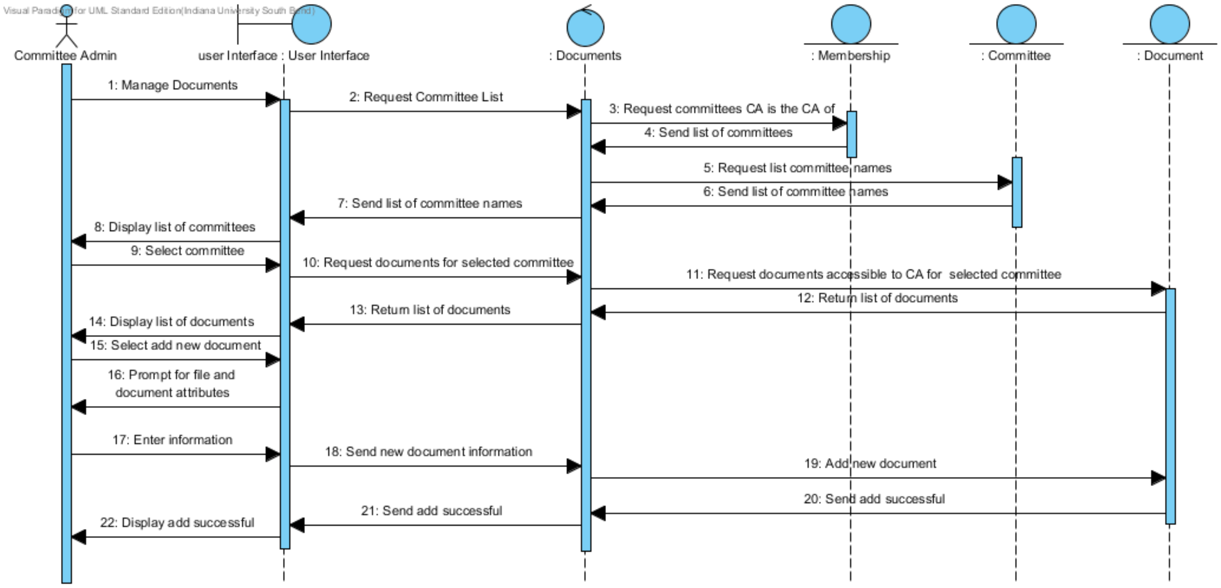
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
Committee Admin
user Interface : User Interface
: Documents
: Membership
: Committee
: Document
1: Manage Documents
2: Request Committee List
3: Request committees CA is the CA of
4: Send list of committees
5: Request list committee names
6: Send list of committee names
7: Send list of committee names
8: Display list of committees
9: Select committee
10: Request documents for selected committee
11: Request documents accessible to CA for selected committee
12: Return list of documents
13: Return list of documents
14: Display list of documents
15: Select add new document
16: Prompt for file and document attributes
17: Enter information
18: Send new document information
19: Add new document
20: Send add successful
21: Send add successful
22: Display add successful

Figure 19 Sequence diagram for Manage Documents scenario: add new document

8.2 Reports

8.2.1 Scenario 1: Display “Committee without enough members report”

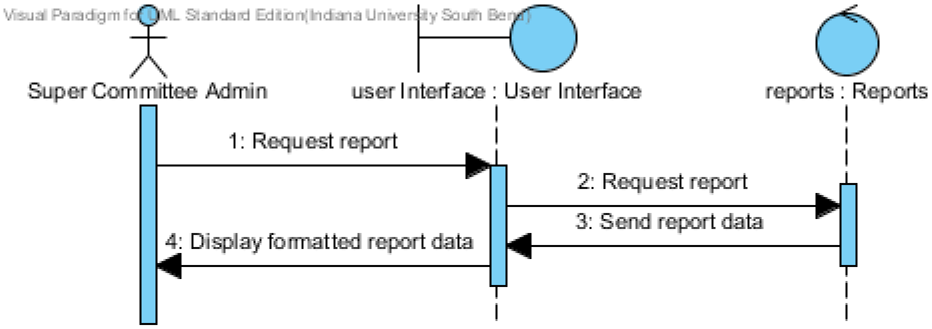
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
Super Committee Admin
user Interface : User Interface
reports : Reports
1: Request report
2: Request report
3: Send report data
4: Display formatted report data

Figure 20 Sequence diagram for Reports scenario

8.3 Manage Meetings

8.3.1 Scenario 1: The CA adds a new meeting

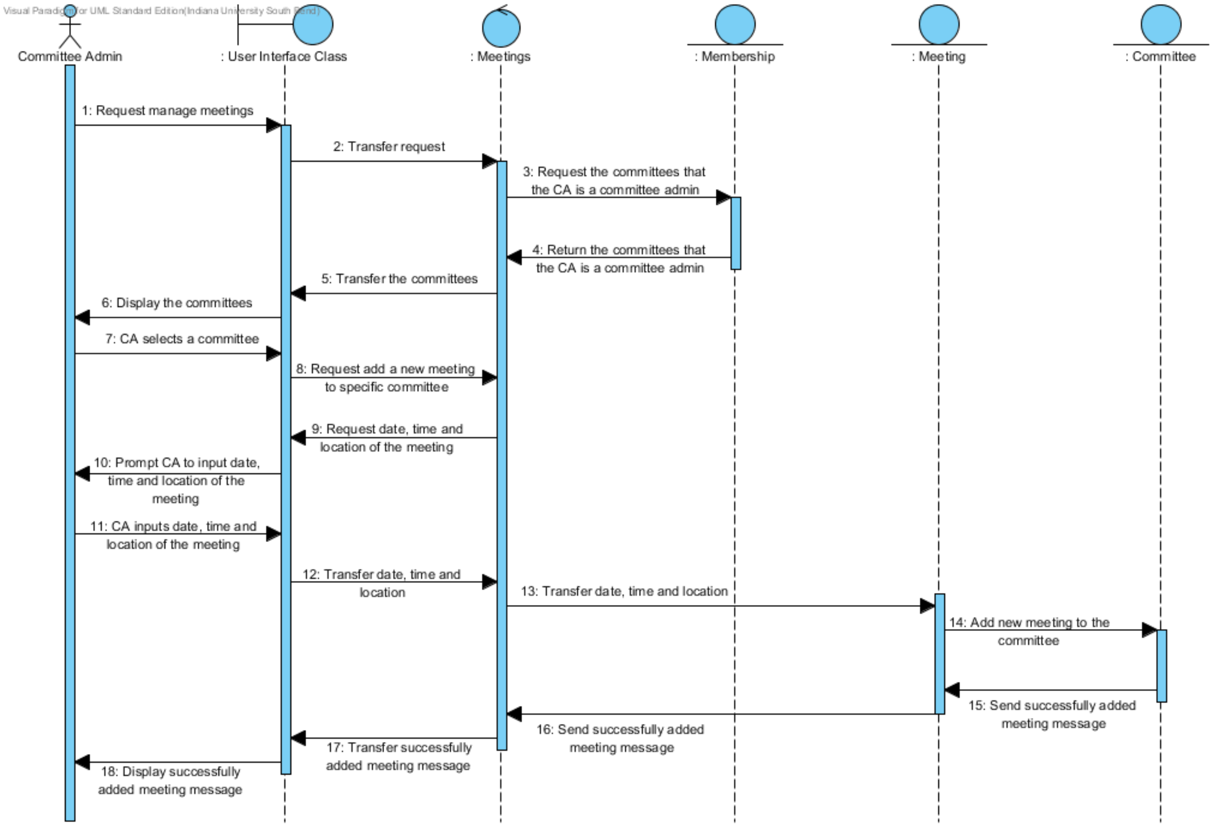
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
Committee Admin
: User Interface Class
: Meetings
: Membership
: Meeting
: Committee
1: Request manage meetings
2: Transfer request
3: Request the committees that the CA is a committee admin
4: Return the committees that the CA is a committee admin
5: Transfer the committees
6: Display the committees
7: CA selects a committee
8: Request add a new meeting to specific committee
9: Request date, time and location of the meeting
10: Prompt CA to input date, time and location of the meeting
11: CA inputs date, time and location of the meeting
12: Transfer date, time and location
13: Transfer date, time and location
14: Add new meeting to the committee
15: Send successfully added meeting message
16: Send successfully added meeting message
17: Transfer successfully added meeting message
18: Display successfully added meeting message

Figure 21 Sequence diagram for Manage Meetings scenario: the CA adds a new meeting

8.3.2 Scenario 2: Convener modifies a meeting

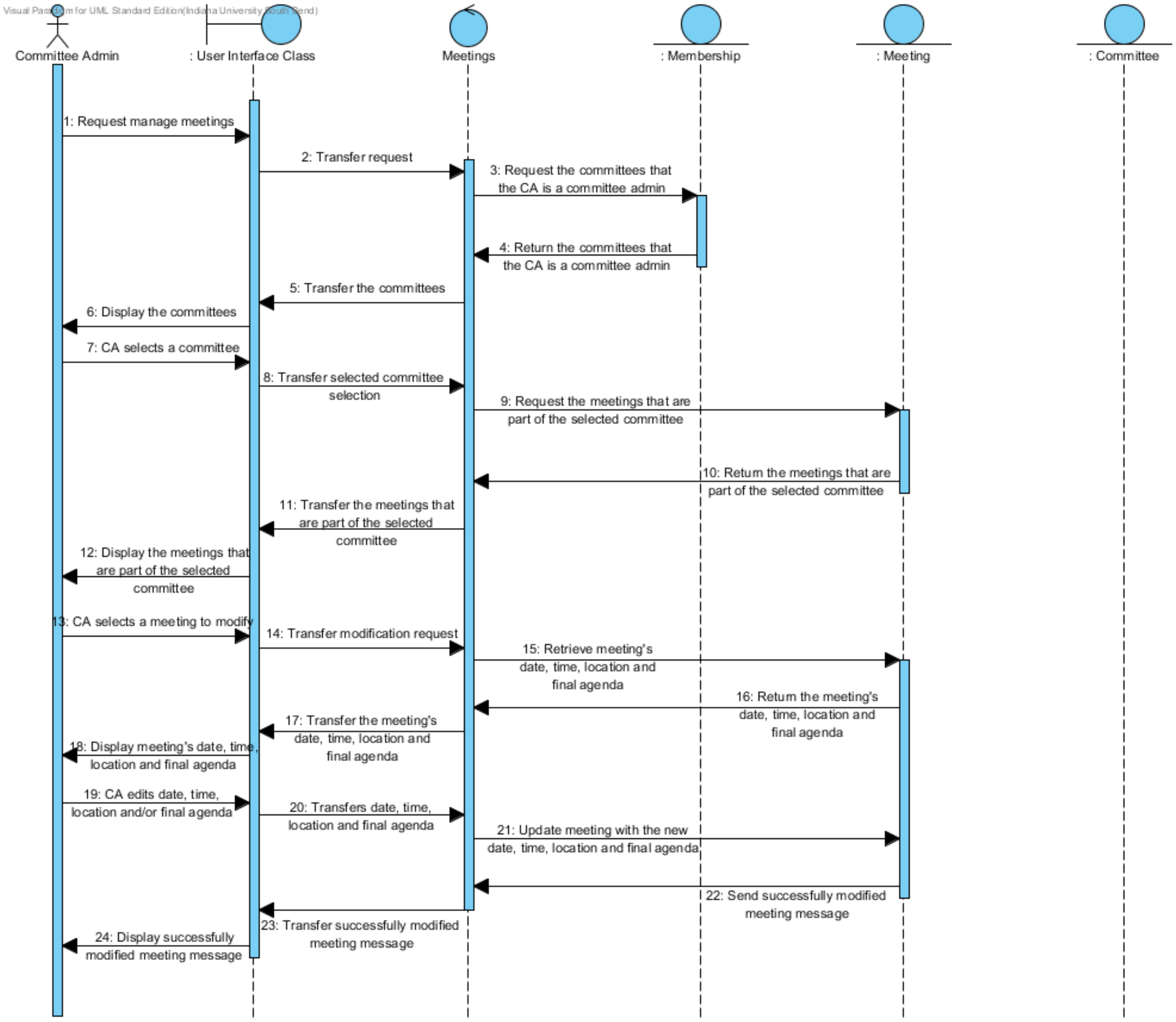
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
Committee Admin
: User Interface Class
Meetings
: Membership
: Meeting
: Committee
1: Request manage meetings
2: Transfer request
3: Request the committees that the CA is a committee admin
4: Return the committees that the CA is a committee admin
5: Transfer the committees
6: Display the committees
7: CA selects a committee
8: Transfer selected committee selection
9: Request the meetings that are part of the selected committee
10: Return the meetings that are part of the selected committee
11: Transfer the meetings that are part of the selected committee
12: Display the meetings that are part of the selected committee
13: CA selects a meeting to modify
14: Transfer modification request
15: Retrieve meeting's date, time, location and final agenda
16: Return the meeting's date, time, location and final agenda
17: Transfer the meeting's date, time, location and final agenda
18: Display meeting's date, time, location and final agenda
19: CA edits date, time, location and/or final agenda
20: Transfers date, time, location and final agenda
21: Update meeting with the new date, time, location and final agenda
22: Send successfully modified meeting message
23: Transfer successfully modified meeting message
24: Display successfully modified meeting message

Figure 22 Sequence diagram for Manage Meetings scenario: Convener Modifies a meeting

8.4 Manage Discussions

8.4.1 Scenario 1: CA adds a new discussion

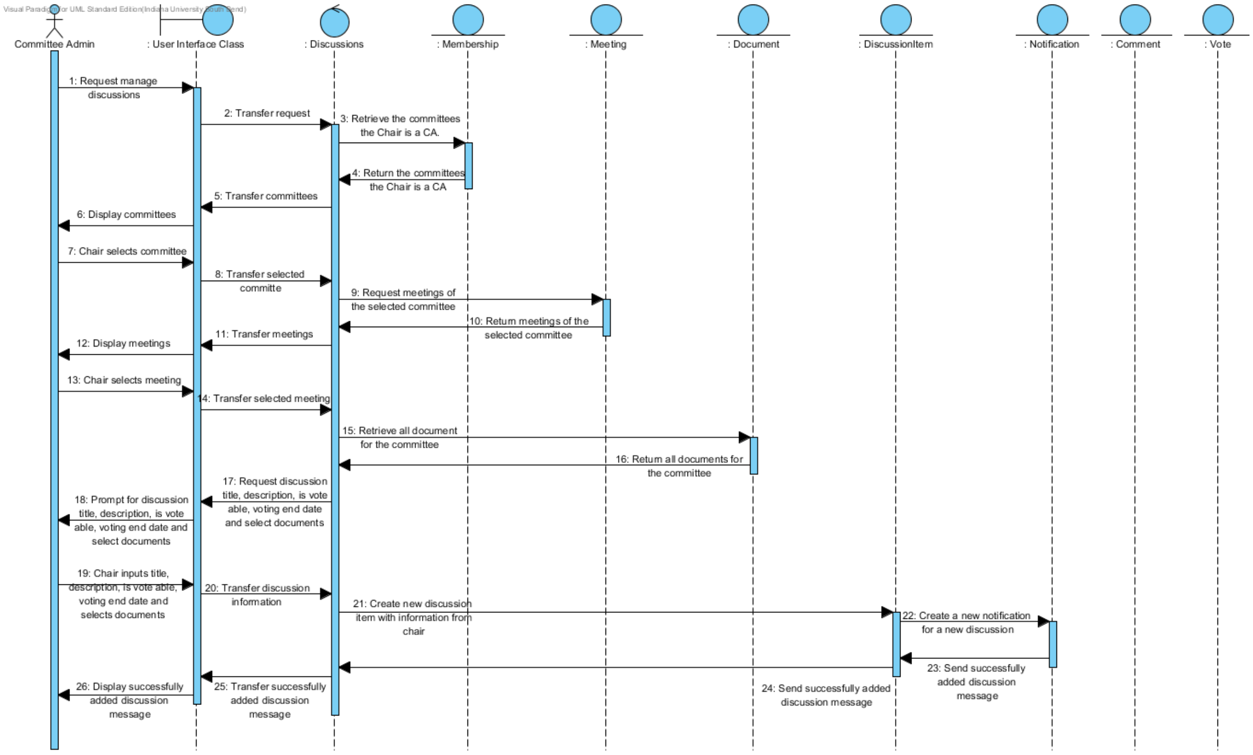
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
Committee Admin
: User Interface Class
: Discussions
: Membership
: Meeting
: Document
: DiscussionItem
: Notification
: Comment
: Vote
1: Request manage discussions
2: Transfer request
3: Retrieve the committees the Chair is a CA.
4: Return the committees the Chair is a CA
5: Transfer committees
6: Display committees
7: Chair selects committee
8: Transfer selected committe
9: Request meetings of the selected committee
10: Return meetings of the selected committee
11: Transfer meetings
12: Display meetings
13: Chair selects meeting
14: Transfer selected meeting
15: Retrieve all document for the committee
16: Return all documents for the committee
17: Request discussion title, description, is vote able, voting end date and select documents
18: Prompt for discussion title, description, is vote able, voting end date and select documents
19: Chair inputs title, description, is vote able, voting end date and selects documents
20: Transfer discussion information
21: Create new discussion item with information from chair
22: Create a new notification for a new discussion
23: Send successfully added discussion message
24: Send successfully added discussion message
25: Transfer successfully added discussion message
26: Display successfully added discussion message

Figure 23 Sequence diagram for Manage Discussions scenario: CA adds a new discussion

8.4.2 Scenario 2: Member adds a comment

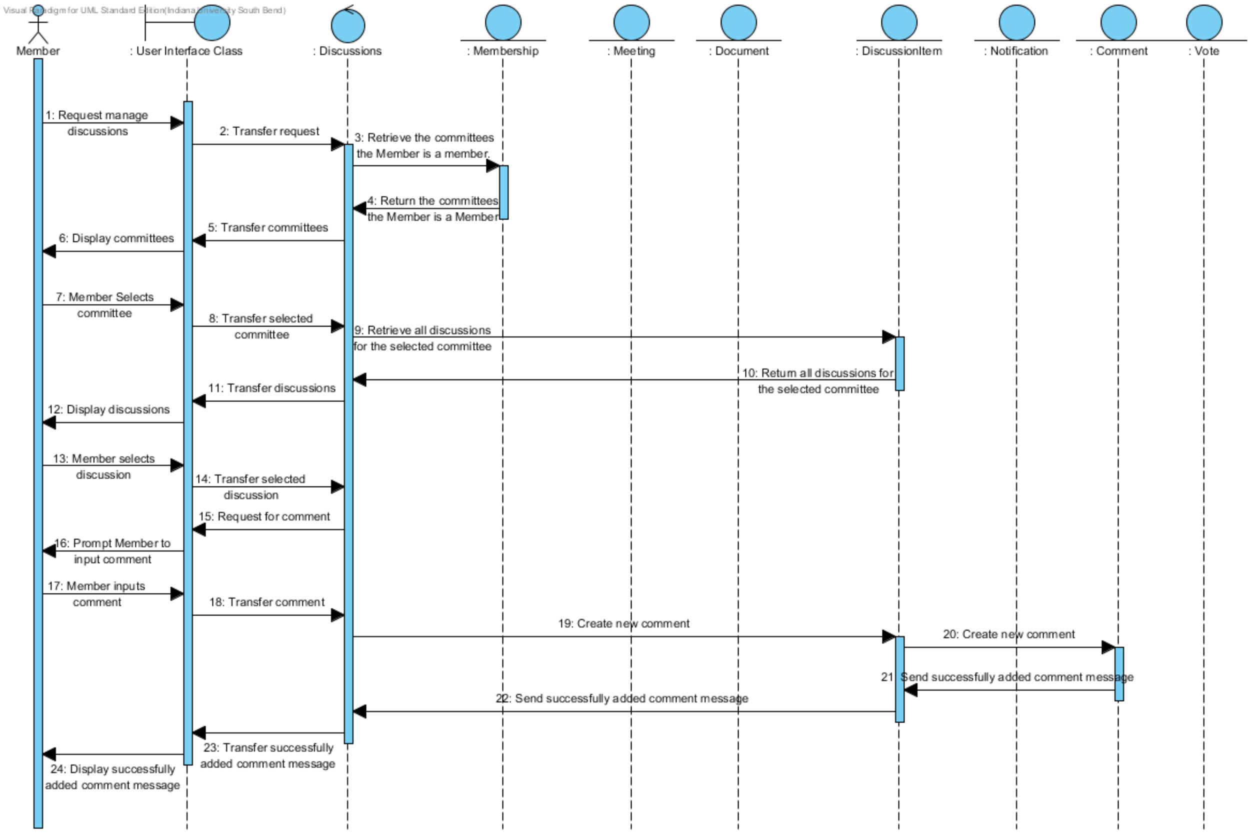
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
Member
: User Interface Class
: Discussions
: Membership
: Meeting
: Document
: DiscussionItem
: Notification
: Comment
: Vote
1: Request manage discussions
2: Transfer request
3: Retrieve the committees the Member is a member.
4: Return the committees the Member is a Member
5: Transfer committees
6: Display committees
7: Member Selects committee
8: Transfer selected committee
9: Retrieve all discussions for the selected committee
10: Return all discussions for the selected committee
11: Transfer discussions
12: Display discussions
13: Member selects discussion
14: Transfer selected discussion
15: Request for comment
16: Prompt Member to input comment
17: Member inputs comment
18: Transfer comment
19: Create new comment
20: Create new comment
21: Send successfully added comment message
22: Send successfully added comment message
23: Transfer successfully added comment message
24: Display successfully added comment message

Figure 24 Sequence diagram for Manage Discussions scenario: member adds a comment

8.5 Search

8.5.1 Scenario 1: Search

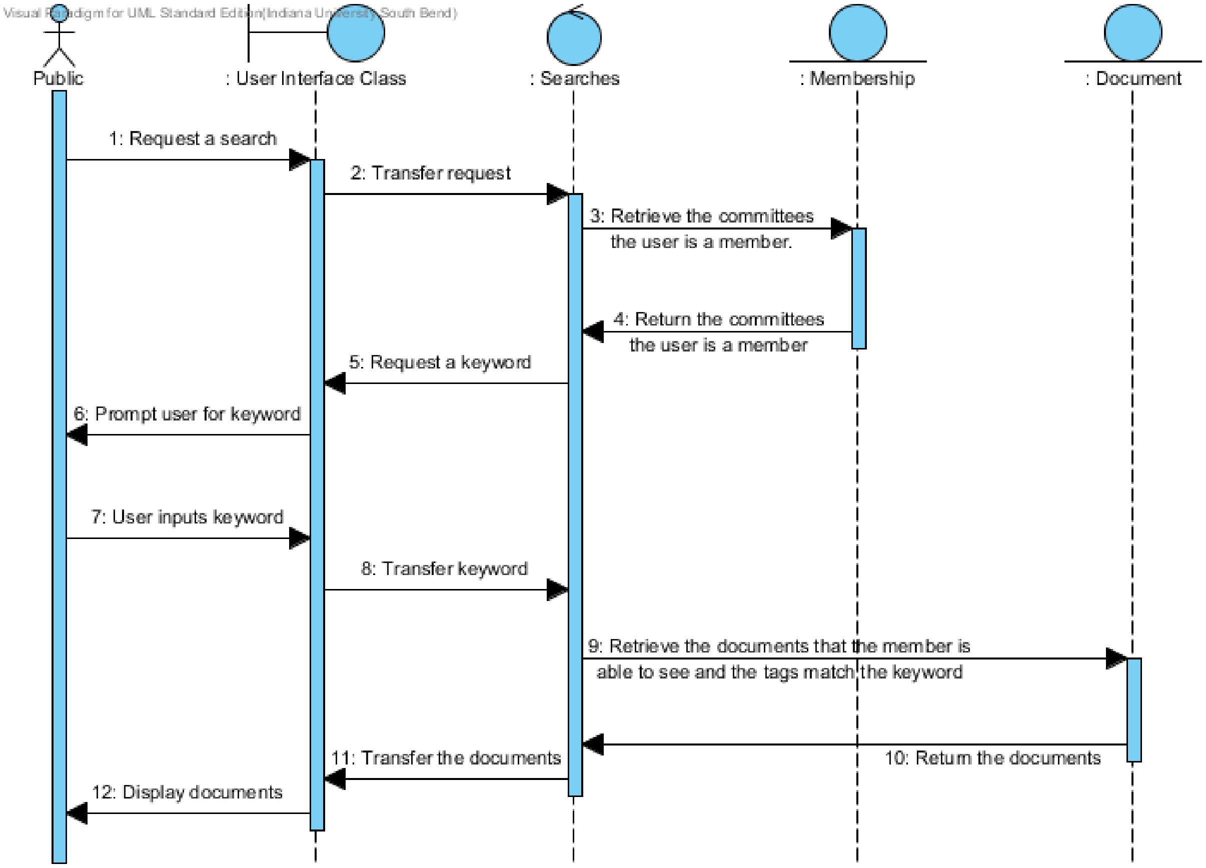
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
Public
: User Interface Class
: Searches
: Membership
: Document
1: Request a search
2: Transfer request
3: Retrieve the committees the user is a member.
4: Return the committees the user is a member
5: Request a keyword
6: Prompt user for keyword
7: User inputs keyword
8: Transfer keyword
9: Retrieve the documents that the member is able to see and the tags match the keyword
10: Return the documents
11: Transfer the documents
12: Display documents

Figure 25 Sequence diagram for Search scenario: search

8.6 Authentication

8.6.1 Scenario 1: Login

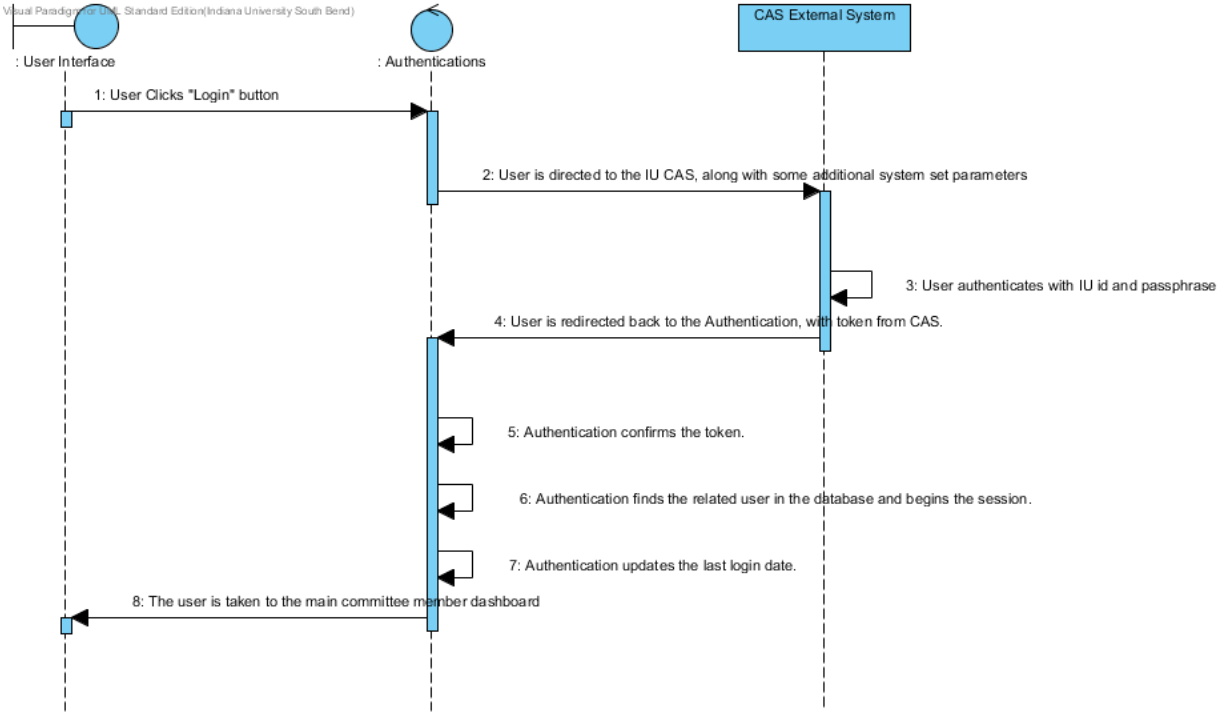
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
: User Interface
: Authentications
CAS External System
1: User Clicks "Login" button
2: User is directed to the IU CAS, along with some additional system set parameters
3: User authenticates with IU id and passphrase
4: User is redirected back to the Authentication, with token from CAS.
5: Authentication confirms the token.
6: Authentication finds the related user in the database and begins the session.
7: Authentication updates the last login date.
8: The user is taken to the main committee member dashboard

Figure 26 Sequence diagram for Authentication scenario: login

8.6.2 Scenario 2: Logout

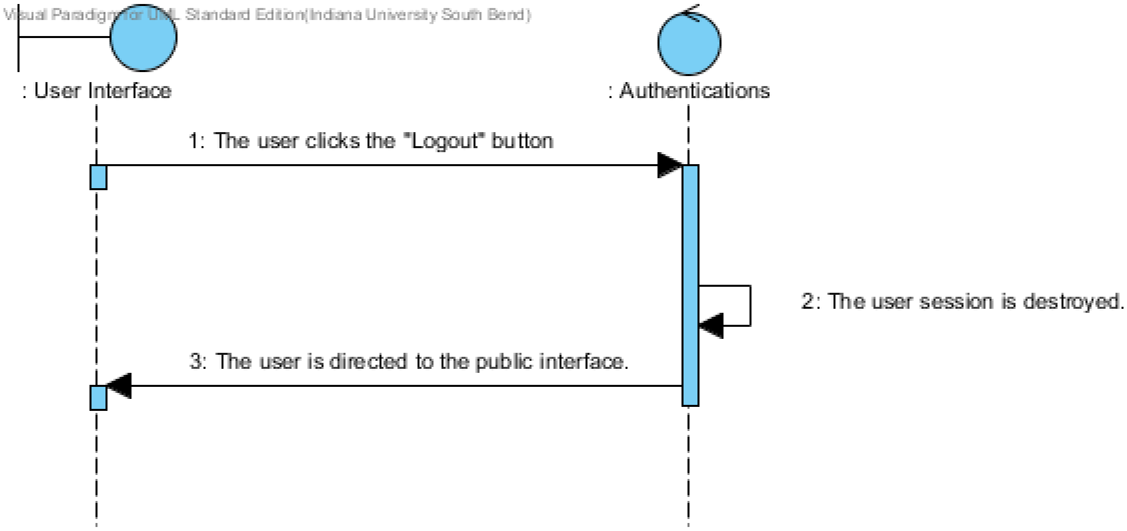
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
: User Interface
: Authentications
1: The user clicks the "Logout" button
2: The user session is destroyed.
3: The user is directed to the public interface.

Figure 27 Sequence diagram for Authentication scenario: logout

8.7 Manage Divisions

8.7.1 Scenario 1: Add Division (Department/Unit)

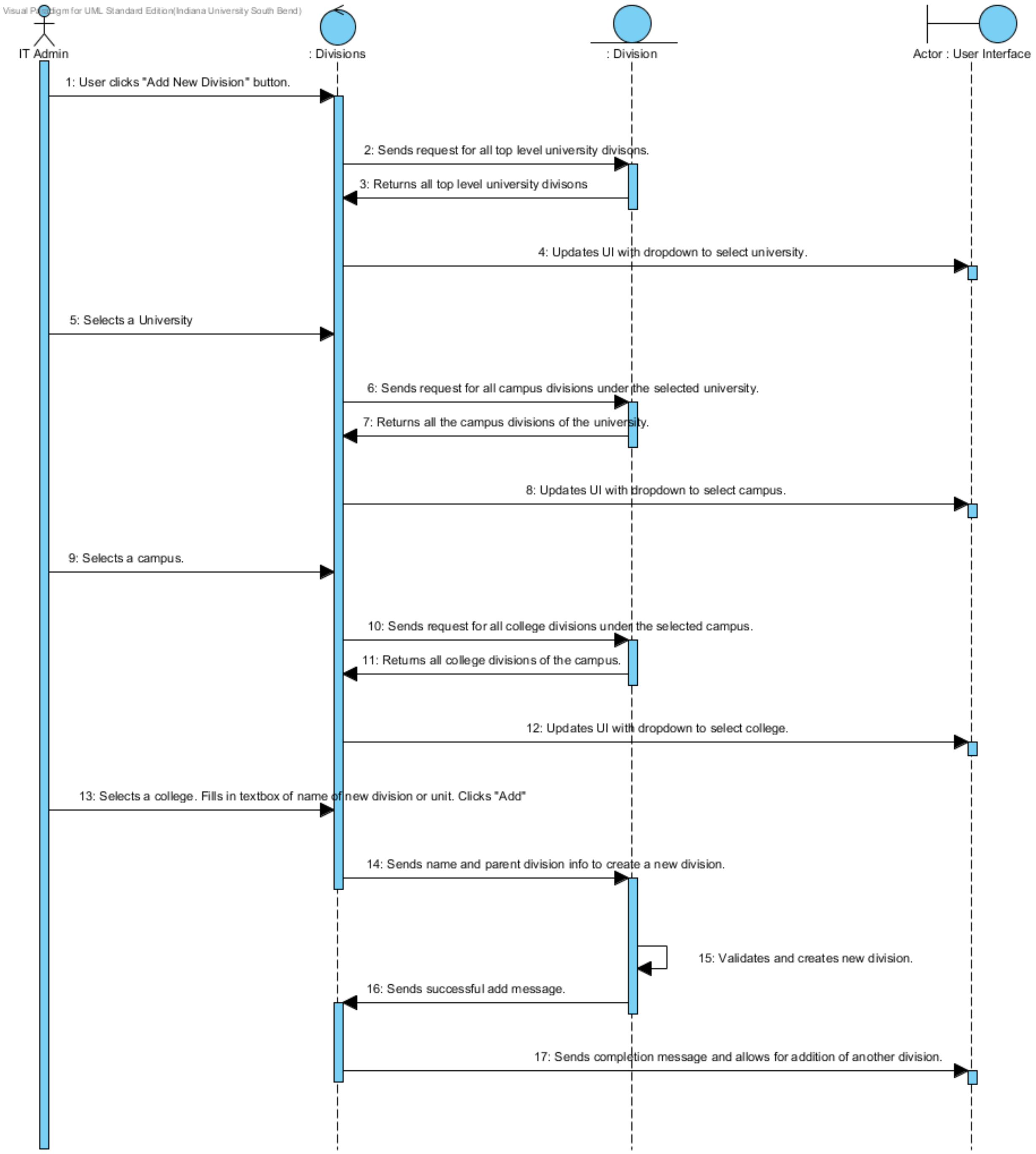
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
IT Admin
: Divisions
: Division
Actor : User Interface
1: User clicks "Add New Division" button.
2: Sends request for all top level university divisons.
3: Returns all top level university divisons
4: Updates UI with dropdown to select university.
5: Selects a University
6: Sends request for all campus divisions under the selected university.
7: Returns all the campus divisions of the university.
8: Updates UI with dropdown to select campus.
9: Selects a campus.
10: Sends request for all college divisions under the selected campus.
11: Returns all college divisions of the campus.
12: Updates UI with dropdown to select college.
13: Selects a college. Fills in textbox of name of new division or unit. Clicks "Add"
14: Sends name and parent division info to create a new division.
15: Validates and creates new division.
16: Sends successful add message.
17: Sends completion message and allows for addition of another division.

Figure 28 Sequence diagram for Manage Division scenario: Add Division

8.7.2 Scenario 2: Modify Division

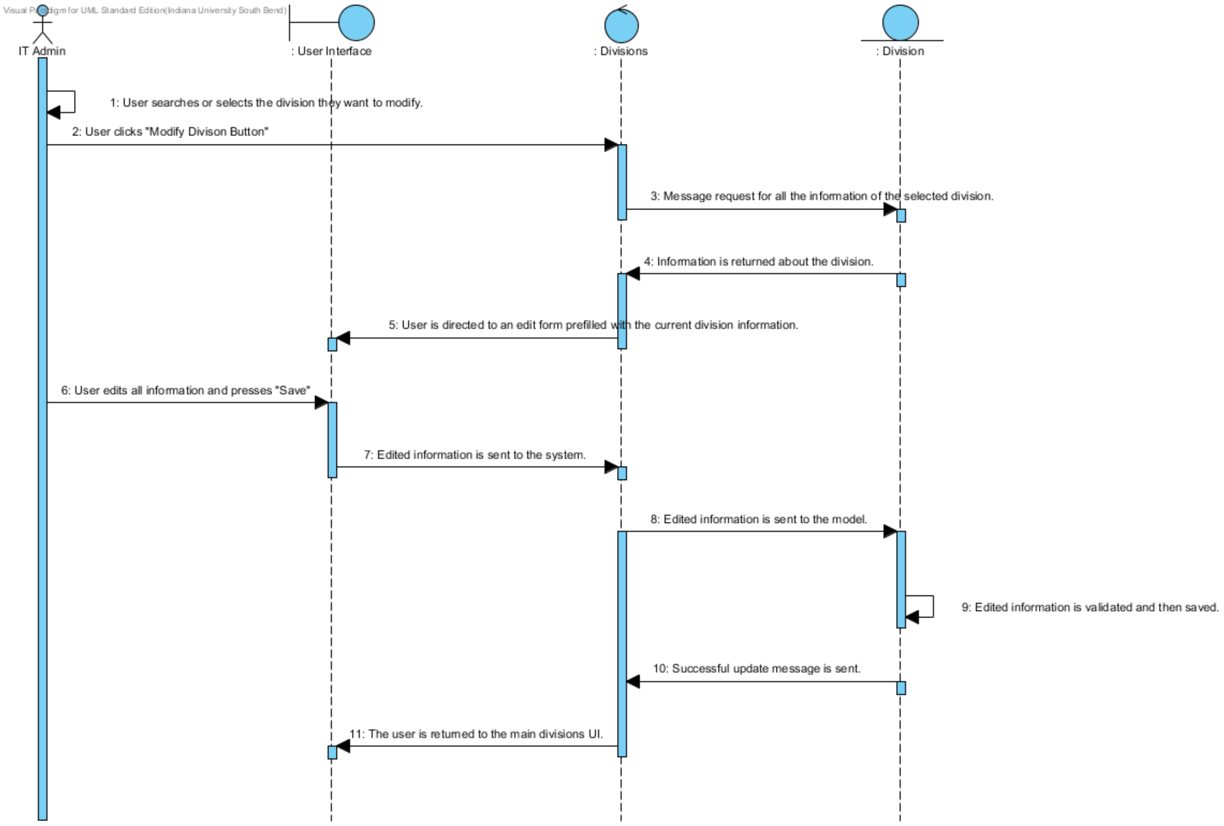
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
IT Admin
: User Interface
: Divisions
: Division
1: User searches or selects the division they want to modify.
2: User clicks "Modify Divison Button"
3: Message request for all the information of the selected division.
4: Information is returned about the division.
5: User is directed to an edit form prefilled with the current division information.
6: User edits all information and presses "Save"
7: Edited information is sent to the system.
8: Edited information is sent to the model.
9: Edited information is validated and then saved.
10: Successful update message is sent.
11: The user is returned to the main divisions UI.

Figure 29 Sequence diagram for Manage Division scenario: modify division

8.7.3 Scenario 3: Delete Division

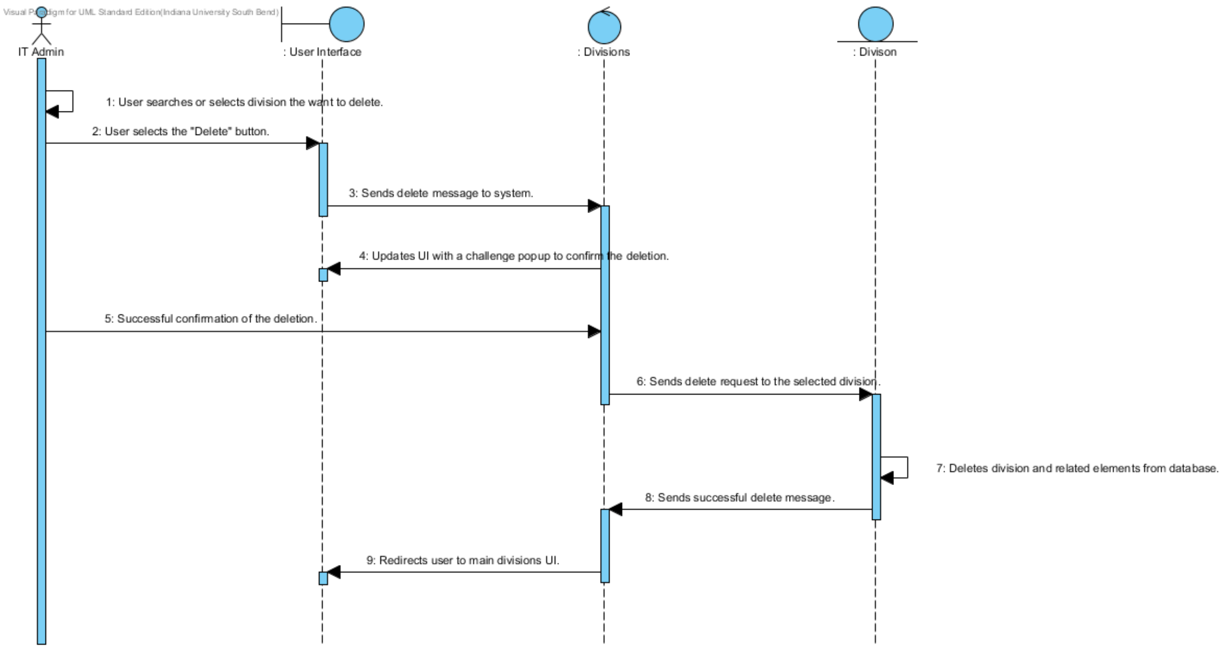
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
IT Admin
: User Interface
: Divisions
: Divison
1: User searches or selects division the want to delete.
2: User selects the "Delete" button.
3: Sends delete message to system.
4: Updates UI with a challenge popup to confirm the deletion.
5: Successful confirmation of the deletion.
6: Sends delete request to the selected division.
7: Deletes division and related elements from database.
8: Sends successful delete message.
9: Redirects user to main divisions UI.

Figure 30 Sequence diagram for Manage Division scenario: delete division

8.8 View Audit Logs

8.8.1 Scenario 1: IT Admin view

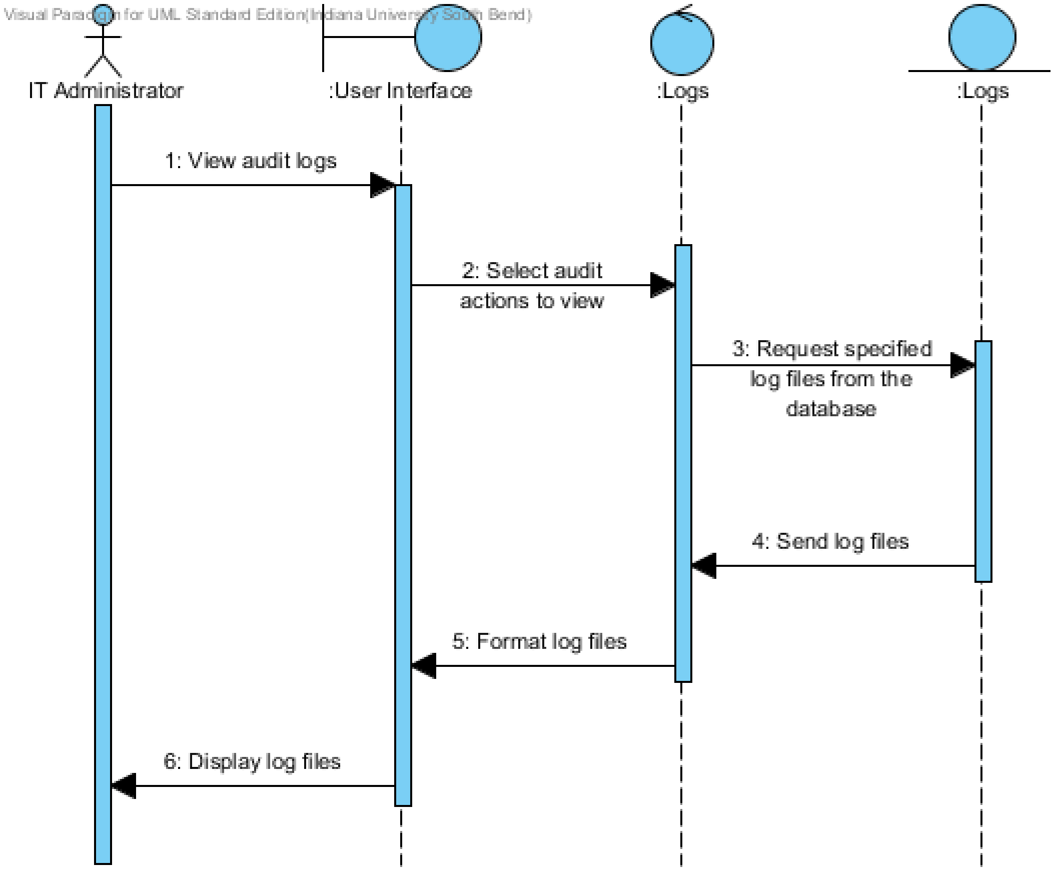
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
IT Administrator
:User Interface
:Logs
:Logs
1: View audit logs
2: Select audit actions to view
3: Request specified log files from the database
4: Send log files
5: Format log files
6: Display log files

Figure 31 Sequence diagram for View Audit Logs scenario: IT admin view

8.8.2 Scenario 2: Committee Admin View

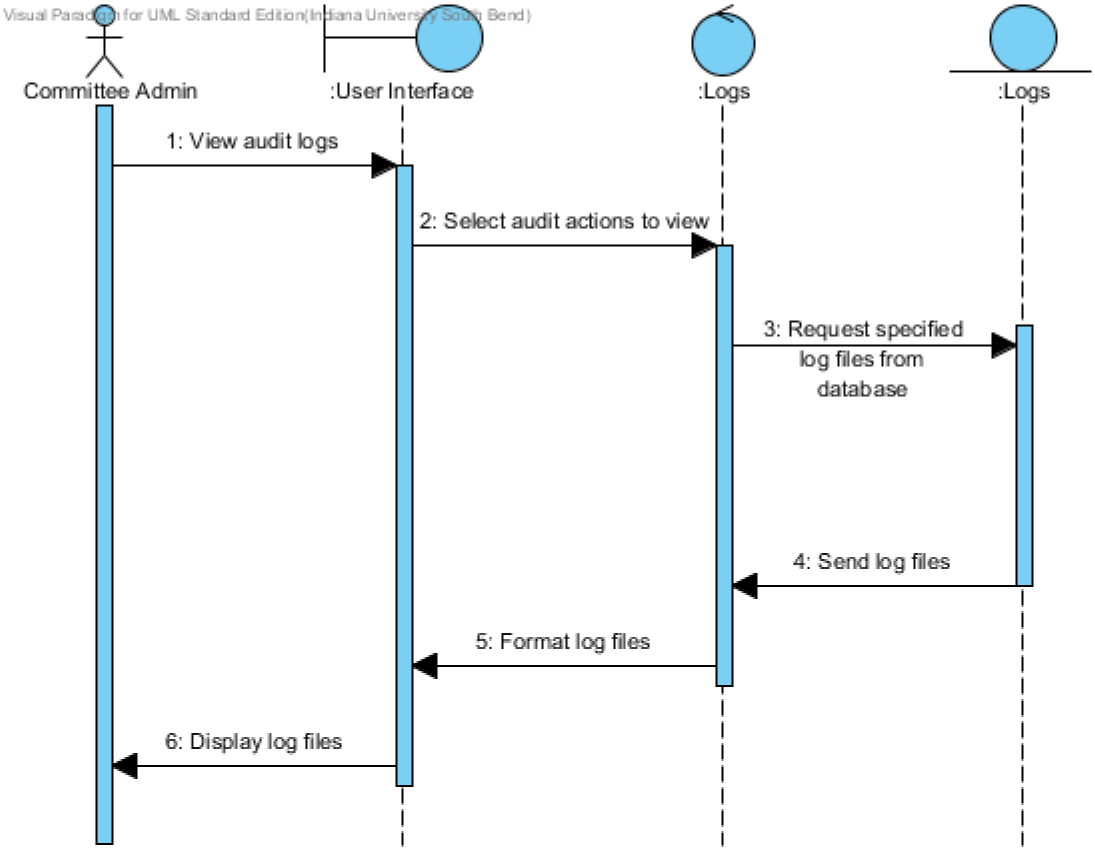
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
Committee Admin
:User Interface
:Logs
:Logs
1: View audit logs
2: Select audit actions to view
3: Request specified log files from database
4: Send log files
5: Format log files
6: Display log files

Figure 32 Sequence diagram for View Audit Logs scenario: committee admin view

8.9 Manage notifications

8.9.1 Scenario 1: Create discussion item

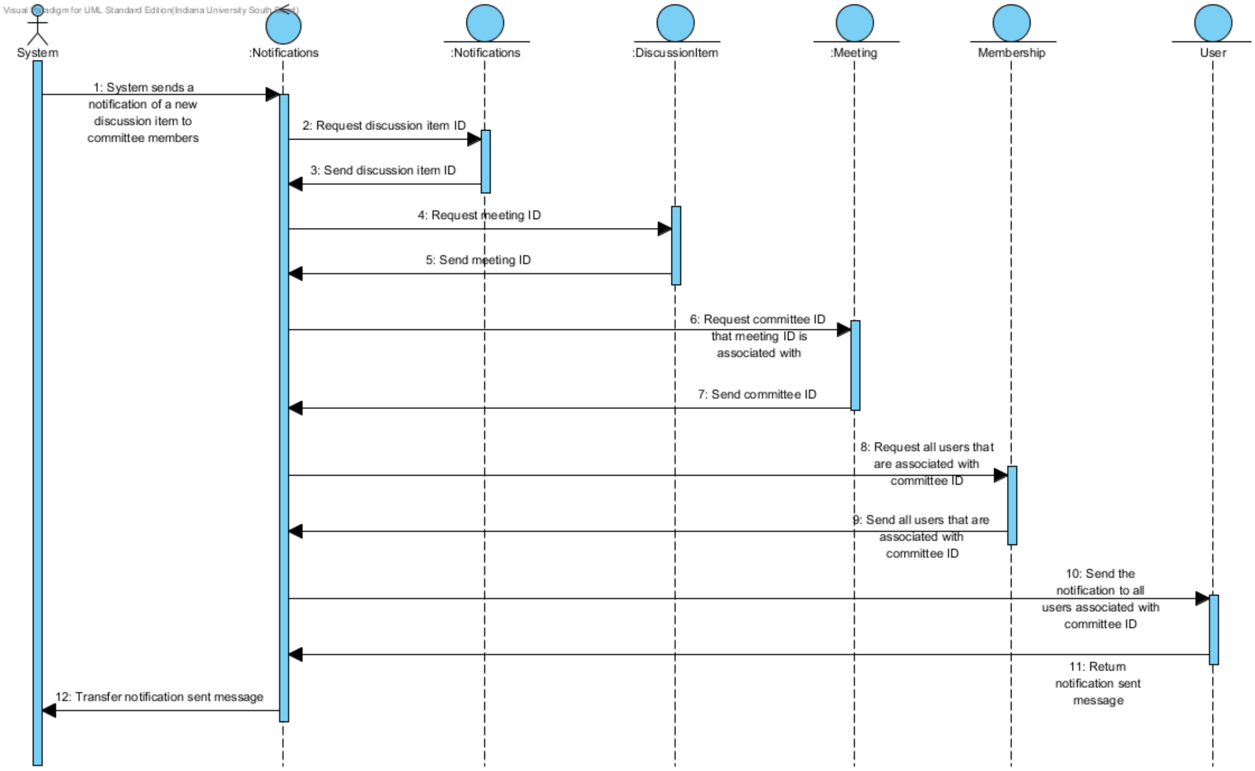
System
:Notifications
:Notifications
:DiscussionItem
:Meeting
Membership
User
1: System sends a notification of a new discussion item to committee members
2: Request discussion item ID
3: Send discussion item ID
4: Request meeting ID
5: Send meeting ID
6: Request committee ID that meeting ID is associated with
7: Send committee ID
8: Request all users that are associated with committee ID
9: Send all users that are associated with committee ID
10: Send the notification to all users associated with committee ID
11: Return notification sent message
12: Transfer notification sent message

Figure 33 Sequence diagram for Manage Notifications scenario: create discussion items

8.10 Manage Committees

8.10.1 Scenario 1: Add Committee

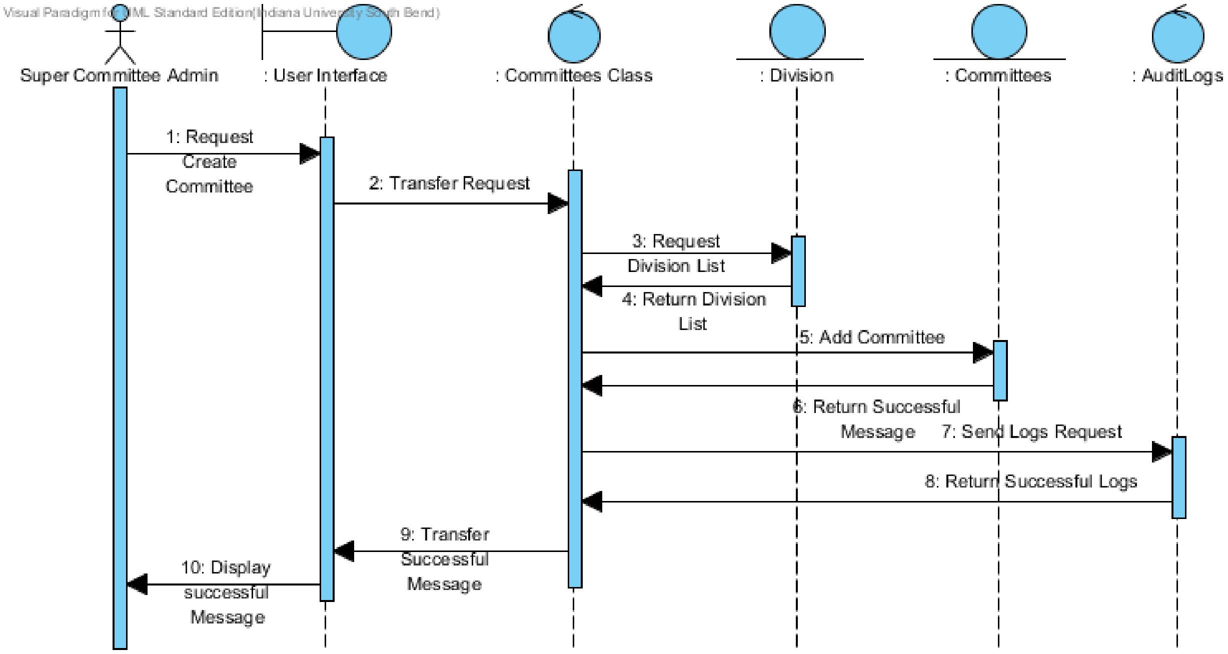
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
Super Committee Admin
: User Interface
: Committees Class
: Division
: Committees
: AuditLogs
1: Request Create Committee
2: Transfer Request
3: Request Division List
4: Return Division List
5: Add Committee
6: Return Successful Message
7: Send Logs Request
8: Return Successful Logs
9: Transfer Successful Message
10: Display successful Message

Figure 34 Sequence diagram for Manage Committees scenario: add committee

8.11 Manage Membership

8.11.1 Scenario 1: Add member to committee

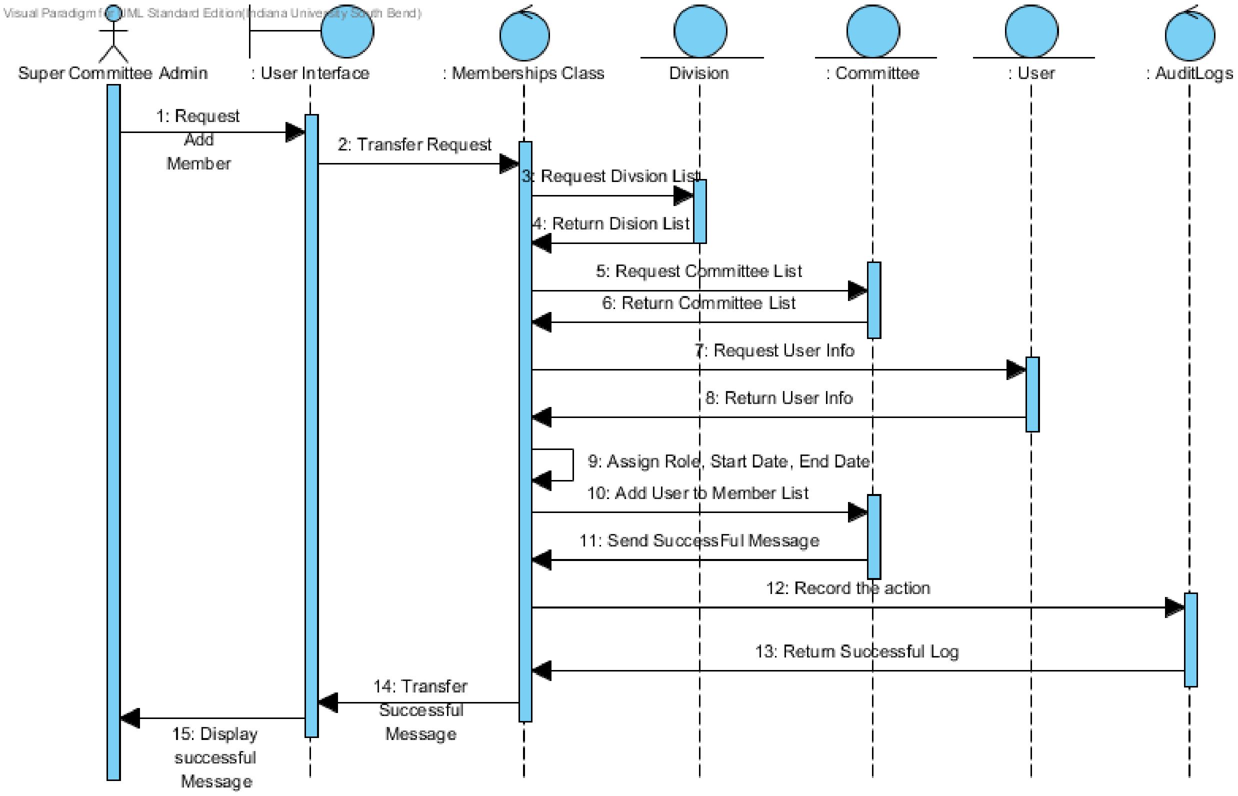
Visual Paradigm for UML Standard Edition(Indiana University South Bend)
Super Committee Admin
: User Interface
: Memberships Class
Division
: Committee
: User
: AuditLogs
1: Request Add Member
2: Transfer Request
3: Request Divsion List
4: Return Dision List
5: Request Committee List
6: Return Committee List
7: Request User Info
8: Return User Info
9: Assign Role, Start Date, End Date
10: Add User to Member List
11: Send SuccessFul Message
12: Record the action
13: Return Successful Log
14: Transfer Successful Message
15: Display successful Message

Figure 35 Sequence diagram for Manage Membership scenario: add member to committee.

9.0 Restrictions, limitations, and other issues

None

10.0 Appendices

None