



# nEdCom

(Web-based Educational Networking Platform for University Students)

Course: CSCI441\_VA\_F2018: 2018F\_CSCI441\_VA\_Software Engineering

Team Name: HKMS

Team Members: Heanh Sok, Kanika Montha, Molika Meas, Sokchamroen Riem

GitHub Link: <https://github.com/heanh sok/hkms-fhsu-se-project>

**Team member contribution**

All team members contributed equally to this project.

<b>nEdCom</b>	<b>1</b>
(Web-based Educational Networking Platform for University Students)	1
1. Customer Statement of Requirement (CSR)	5
2. Glossary of Terms	8
3. System Requirements / User's story	9
3.1. Enumerated Functional Requirement	9
3.2. Enumerated Non-functional Requirement	11
3.3. On-Screen Appearance Requirement	12
4. Functional Requirements Specification	14
4.1. Stakeholders	14
4.2. Actors and Goals	14
4.3. Use Cases	15
4.3.a. Casual Description	15
4.3.b. Use Case Diagram	17
4.3.c. Traceability Matrix	19
4.3.d. Fully-Dressed Description	22
4.4. System Sequence Diagrams	30
4.5. User Effort Estimation	46
5. Effort Estimation using Use Case Points	47
6. Domain Analysis	53
1. Concept Definition	53
2. Association Definition	54
3. Attribute Definitions	57
7. Interaction Diagram	59
8. Class Diagrams And Interface Specification	68
a. Class Diagram	68
b. Data Types And Operation Signatures	69
c. Traceability Matrix	71
9. System Architecture And System Design	72
a. Architectural Styles	72
b. Identifying Subsystems	73
c. Mapping Subsystems to Hardware	74
d. Persistent Data Storage	74
e. Network Protocol	75
f. Global Control Flow	76
g. Hardware Requirements	76
10. Algorithm And Data Structures	77
11. User Interface Design And Implementation	78

12. Design of Tests	83
13. Project Management	89
a. History of Work, Current Status, and Future Work	89
b. References	90

## 1. Customer Statement of Requirement (CSR)

Even though there are a lot of platforms for chatting, networking, sharing of information, but there is no clear way for students in Cambodia to network academically and professionally. Students often follow pages on social media platform such as Facebook to get scholarships or school information. But they may get distracted by contents of other posts that appear in the newsfeed or chat notifications from friends. Students may need to ask information regarding a school program that they're interested in but not know any other students who are willing to provide the assistance or they might find it difficult to reach out to others to ask questions about careers or educational plans. Apparently, those who are experienced in professional development have the capability to help addressing the questions and concerns of their peers regarding career development, school programs, and other opportunities or might as well be able to provide mentorship. A new implementation is needed in order to bridge this networking gap and gather all professional development areas in one place.

A platform which enables the enhancement of higher education is highly in need while Cambodia consists of thousands young people which shares more half of the total population. This is the key issue that developing countries like Cambodia need to focus on. Higher education plays an important role to bring up more competitive workforce in the future especially those with ready-skill for the real workplace.

Although strengthening curriculum is not enough, we need more modernising infrastructure with tech-integrated solution. Youths require more skill to remain employed in this competitive and rigorous labor market. According to East Asia Forum, “Cambodia should enact policies to stimulate technological development and promote collaboration in research and development activities.” (East Asia Forum, 2014)

*“Growing labor force alone is not sufficient to attract and create jobs. Here, employers particularly face significant skill shortages, so education system and the increase of investment in education has a role to play in filling those gaps. But if some part of the challenges can be addressed, looking at the alternative resources, modernizing education and vocational training, stepping in skill developments provided by the firms are main direction for labor forces to create future jobs,” Inguna Dobrja, the World Bank’s country manager in Cambodia, said at the launching of the report. (World Bank in Cambodia )*

Insufficient skilled labors creates skill gap which is caused by the shortages of informed information, advice and clear guidances. However, about 45 percent of the population have access to the internet. This means internet infrastructure is available for more digital platform to operate and more students will be able to access more information online.

With a vision to solve employment issue by establishing an encouraging platform, our proposed solution is to create a educational networking platform to help and prepare Cambodian students in the job

market, find mentorship, explore many opportunities, possible career paths, and move ahead in their educational pathway such as getting scholarships, building professional skills, etc.

### What Students in Higher Education in Cambodia Need.

- a. More information about educational curriculum and other academic topics
- b. Mentorship for advice and guidance.
- c. A well-rounded and trustworthy information about different opportunities in society
- d. Career matching and professional training to be ready for workplaces after graduation

The purpose of our application is to serve as a sharing platform that connect students from all educational and professional background to contribute information about schools, university majors, admission application processes, job skills, etc. On top of that, event organizers, employers, recruiters, or even faculties can contribute to sharing information, assisting, and mentoring other students.and can post information about jobs, scholarship opportunities or course related information in their programs. Companies can post job announcement and search for potential employees through our platform, which will make the searching process easier.

There are pages like [Wedushare](#) where students can look for scholarship programs and job opportunities, [LinkedIn](#) where users build their network and professional portfolio that ease with job search and recruitment process, and [Barcamp](#) events in Cambodia that gathers innovators, professionals, enterprises from big to small, educators, and students from diverse background to come together and share their knowledge, present their product or idea to the public, and exchange experience and skills. However, they're not a zone that focuses on bringing student on a complete educational and professional development experience through digital networking.

To address the issue mentioned above, we will create a web-based application with the following functionalities:

- **Forum** where users can ask questions, exchange ideas, initiate discussion topics related to the fields they're interested in. They can tag the question in the appropriate categories so that other users can easily look for the posts through the filter tag. Other users can participate in the forum post by making comments, upvote, or downvote the post to verify whether the post is relevant, useful or helpful.
- **Opportunity Page** where companies can post job or internship announcements, event organizers can post about their events or event volunteer opportunity, and universities can post about their university programs, scholarships, seminars and other information useful for prospective students. Many educational, inspirational events and activities are being conducted in many different places regarding to many interesting topics. With this platform, students will be more aware and more informed of nearby events and other interesting events conducted by other universities. Students can network and address date to join those interesting events. Students will also be notified with other opportunities such as scholarships, internship, training and exchange program.

- **User profile page** where users can upload their profile picture, write information about themselves such as education background, work experience and expertise and more elaborate description. They can modify the setting to hide or display certain information from their profile page. The profile page also shows user post histories in forums and their rating in the system. Users can also follow others in order to get notifications related to posts, news, etc.
- **Rating system** which is displayed in the form of points. Users who contribute a lot in forums and get a lot of upvotes will gain high points. Users who achieve certain point threshold will receive a certain status name.
- **User roles** which determines accessibility of the system to different user roles. Thus, on the user sign up page, the system will enable user to register themselves as standard user or premium user. Standard user can be students or anyone. Premium users are important contributors such as faculties, event organizer, recruiter, etc. The registration of premium user is to be approved by the system administrator in order for the account to be created.
- **Private chat and email** function where users can reach out to others. It's important for system to have chat templates for users such as "Hello, I would like to pair up with you on nEdCom." By having system-generated casual phrases as chat templates, users shall be able to break the ice with other users easier, therefore it increases the amount of interaction between users.
- **Mentorship** a function for student to be matched with academic mentor who can be professor or senior students from different university. In this section students can interact with professional and seek for advice or guidance regarding to their university life. More importantly, this function also serves as a platform for career mentorship.

## 2. Glossary of Terms

- **nEdCom:** Short for Networking, Education, and Communication. It's a web-based application which serves as a platform to help prepare students for higher education and career development while expanding the network.
- **Points (Rating):** A feature which allows students to rate other people's answers in the forum. The system will average the points the person has accumulated and it will be displayed publicly.
- **Admin Users:** nEdCom team responsible for maintaining the system, moderating posts, and providing support.
- **Standard Users:** Student users, who are non-premium users.
- **Premium User:** Advisors, event organizers, employers, recruiters, or faculties who can contribute to sharing information, assisting, and mentoring other students. To be a premium user, they have to self-register as a premium user and wait for confirmation from administrator.
- **Forum:** A page where users can contribute through posting and sharing information, ask questions, and rate the answers.
- **Discussion Group:** group created specially to for user to discuss about certain topic based on their interest
- **Upvote:** vote for users' posts, questions, or answers to indicate to that they are good, correct, useful, or relevant
- **Downvote:** vote against, users' posts, questions, or answers to indicate that they are inappropriate, false, incorrect, or irrelevant
- **User Profile:** A page for displaying user information.
- **Opportunity Pages:** a special page of the system that displays posts related to events, competitions, scholarships, internships, career, or volunteering opportunities.
- **UX/UI:** Front-end interface design.
- **Top Achiever Page:** the top users with the highest points will be also appear in "Top Achievers" list for public showcase in the platform
- **Central Database:** Remote electronic file system used for storing and retrieving of all nEdCom data
- **Credential:** Authentication of user identity in order to login to use the system.
- **Notification:** Message delivered to user regarding announcement, important updates, and activities related to user account.
- **CAPTCHA:** Computer program used to tell human and machine apart and to prevent malicious activities on the website.

### 3. System Requirements / User's story

#### 3.1. Enumerated Functional Requirement

The following table lists down the requirements of that system-to-be. Each row contains a unique identifier, Priority weight, and the detail of the requirement. The priority weight ranges from 1 to 5 in which 5 is the most essential to the success of the project. The key word “shall” is used to identify the requirement that are essential to implement whereas the key word “should” is used to identify the requirements that are desirable but not mandatory.

Identifier	Priority	Requirement
REQ-1	5	The system shall support registering new standard user with standard permission level (see REQ-6 and REQ-7). The registration for this type of user can be done automatically without waiting for approval from admin. Required information shall include a unique username and a password that conforms to guideline as well as the users' email address, first and last name.
REQ-2	5	The system shall support registering new premium user (specially designed for Advisors, Event Organizers, Employers, Recruiters, Faculties) with special permission level (see REQ-6, REQ-7, and REQ-8). The registration needs verification from admin. Required information shall include a unique username and a password that conforms to guideline as well as the users' email address, first and last name, phone number, and photo of national ID. The system shall alert admin to verify the user while the user waits confirmation.
REQ-3	4	Upon successful registration, the system shall alert the user to add additional information to the profile including: personal information, education background, careers, interests, and projects.
REQ-4	3	The system shall allow users to edit their profile information as needed.
REQ-5	5	The system shall award points to user if any posts/questions/answers of that user get upvoted by other users. In contrast, the system shall deduct points from user if any posts/questions/answers of that user get downvoted by other users.
REQ-6	5	The system shall enable any type of user (standard and premium) to join general forums or discussion groups, ask and answer question, downvote or upvote questions or answers.
REQ-7	5	The system shall enable any type of user (standard or premium) to browse any general opportunity pages such as Events, Competitions, Scholarships, Internships, Career, or Volunteer.

REQ-8	3	The system shall enable any type of user (standard or premium) to bookmark any posts in any opportunity pages.
REQ-9	4	The system shall allow premium user to create forums or discussion group, and post about events, competitions, scholarship, internship, or careers in opportunity page.
REQ-10	4	The system shall allow the user to manage the forums, groups, or posts on any opportunity pages that they have created.
REQ-11	3	The system shall enable user with certain number points to access certain exclusive forums, groups, and opportunity pages.
REQ-12	3	The system shall display the name and the link to profile of user with the highest points to the monthly “Top Achievers” page.
REQ-13	1	The system should disconnect the user session if the user is inactive for a predetermined period.
REQ-14	5	The system shall store and retrieve all the data from a central database.
REQ-15	2	The system should enable user to send private message to one another.
REQ-16	2	The system should allow one user to follow other users on the platform to get notification of their activities.
REQ-17	2	The system should allow mistakes while entering login credential. Nevertheless, to avoid individual with bad intention to brute force the password, the number of failed attempted should be limited to 5. If the number of fail attempt reaches its limit, the system will lock the account and notify the user of this suspicious activity through email.
REQ-18	3	The system shall allow user to reset password by sending a reset password link to the user email. Once the user has opened the link from their email, the system shall ask the user to input a new password.
REQ-19	5	The system shall allow admin with special privilege to oversee and manage all the users, forums, groups, or opportunity pages.
REQ-20	4	The system shall allow the users to search for relevant informations on the platform with certain keywords.
REQ-32	5	The system shall allow the users to view, add ,and edit their profile information including profile photo, personal information, educational background, work experiences, volunteer experiences.
REQ-33	4	The system shall allow admin to manage all user account and their profile information including profile photo, personal information, educational background, work experiences, volunteer experiences.

### 3.2. Enumerated Non-functional Requirement

The nonfunctional requirements are based on FURPS+ – Functionality, Usability, Reliability, Performance, Supportability, and other various attributes.

Identifier	Priority	Requirement
REQ-21	4	The system shall have a clean and simple interface that enable the user to quickly and easily navigate to different pages across the platform.
REQ-22	3	The interface should include different color combinations, icons, and shapes to attract the user attention and help them understand the feature of each page.
REQ-23	4	The system shall have the capacity to handle the incoming traffic when large number of users access the website at the same time.
REQ-24	2	The system should use CAPTCHA to block spammers or bots
REQ-25	3	The system should quickly return search result that relevantly matches the user's specified key words.
REQ-26	4	The user interface shall be responsive such that it will change dynamically with different screen size.
REQ-27	5	The system shall have 24/7 availability for user to access.
REQ-28	3	The web platform should be compatibility with different browsers such as Google Chrome, Mozilla Firefox, Safari, Microsoft Edge, or Opera.
REQ-29	4	The platform shall support different types of devices such Laptop computer, Desktop computer, Smartphone, Tablet that run internet browser.
REQ-30	4	The web platforms shall load or load any pages quickly when the user hits enter or refresh.
REQ-31	4	The system shall have a backup of the central database in case the system crashes or get hacked.

### 3.3. On-Screen Appearance Requirement

In order to help nEdCom users to fully and visually understand how the website works, wireframes are provided in this section. The arrangement of each features or elements as well as the display are not yet finalized and is subject to change. nEdCom website will be responsive in which it can

adapt to the size of the screen a user is viewing the site on. Do note that everything that is provided here is just to give the user the essential information they need in order to understand the necessary features that are available on the website.

	Priority Weight	Description
OSA-1	5	<p><b>Standard Default:</b></p> <p>Users can find elements like menu, logo, search bar, account icon, notification icon in the header of the website. nEdCom logo will locate on the left and will be on display on every webpage. The global search bar will be placed in the center of the header, next to the logo. Navigation menu will be responsive, which means users can choose to fully show or hide, and will locate on the left of the logo. At the bottom of the website, users can find useful links such as About, Contact Us, Privacy Policy and Terms of Service.</p>
OSA-2	5	<p><b>Homepage:</b></p> <p>Homepage will be linked with nEdCom logo. It will divided into 4 sections. The first section, on top of the page, will display featured announcements that nEdCom wants to notify users. New career opportunities and other announcements will be available in section 2. Section 3 displays all available forum topics as well as the new added forums. Testimonials from students, faculty, and recruiters will be added to section 4. This is subject to change for non-login users and login users.</p>
OSA-3	4	<p><b>Student Dashboard:</b></p> <p>Profile customization will be available. Students can build their cv/resume for employment opportunities. They can see all of their posts with total views, votes (up or down), and comments/replies.</p>
OSA-4	4	<p><b>Faculty Dashboard:</b></p> <p>Profile customization will be available. Faculty can see all of their posts with total views, votes (up or down), and comments/replies. They can make a direct post from their account.</p>
OSA-5	4	<p><b>Recruiter Dashboard:</b></p> <p>Profile customization will be available. Recruiters can see all of their posts with total views, votes (up or down), and comments/replies. They can make a direct post from their account.</p>
OSA-6	5	<p><b>Forum:</b></p> <p>Displays the categories of the forums by subject and topics. New forums will be displayed with date, number of views, number of comments.</p>
OSA-7	3	<p><b>Career Page:</b></p>

		Career and internship availabilities will be posted here. Students can search for jobs based on its categories and types (full time, part time or internship).
OSA-9	2	<p><b>Announcement Page:</b>            Events, Scholarships, Competitions, Workshops that open for students regardless of their universities will be available here. Each types of announcement will be categorized in different sections.</p>
OSA-10	1	<p><b>Notification:</b>            This will be accessible on every webpage. Users can see notifications at the top right of the website, next to user account icon.</p>

## 4. Functional Requirements Specification

### 4.1. Stakeholders

Our system is a platform for networking, academic and professional development, therefore it may be of interest to:

- Standard Users: (those who make use of the system for personal and professional growth.)
  - + Students (highschool, university, ...)
  - + Job Seekers
- Premium Users (those who can contribute to sharing and providing information and are able to offer mentorship to other users)
  - + Teachers
  - + Tutors
  - + Recruiters
  - + Administrator
- Companies (Seeking to advertise job announcement and looking for potential employees)
- Universities (Seeking to attract potential students)

### 4.2. Actors and Goals

Actor	Goal
Administrator	To maintain the system, facilitate posts, forum, and ensure smooth operation of application
Standard User (Students / Job Seekers)	To search for career/academic opportunities, seek mentorship, upscale their portfolio, and expand their network.
Premium User (Teachers / Advisors / Tutors / Recruiters)	To post information related to job announcement, scholarships, internship, networking events, and other opportunities, as well as to provide mentorship, help answer questions related to their field, and provide useful information
Web Page	To display application functions and interact with users
Database	To store and enable retrieval of data
Email System	To send notification to users regarding account activities

### 4.3. Use Cases

#### 4.3.a. Casual Description

<b>Use Case</b>	<b>Action</b>	<b>Description</b>
UC-1	Register Standard User	Upon new account registration, user can choose to sign up as standard user. User will be asked to fill in credential informations such as first name, last name, email address, username, and password.
UC-2	Register Premium User	Faculties, recruiters, event organizers, employers sign up for a premium account. To do so users are required to provide their first name, last name, email address, phone number, photo of national ID and proof of employment. The registration requires administrator approval.
UC-3	Account Login	Users fill in their username and password to login, and it will direct them to their profile page.
UC-4	Edit Profile	Users edit their profile information.
UC-5	Post In Forum	Users post any topic/questions to forum.
UC-6	View Forum	Users view forum questions, comments, and number of up/downvotes.
UC-7	Create Discussion Group	Users create discussion group in order to post information/questions specific to that topic. When creating the group, they have to select its category and make it either public or private. After it's created, it will generate a link that directs you to the discussion page which the admin of the group can share with others. If it's a private discussion, it will require permission from the admin before other user can access it.
UC-8	Join Forum Discussion	Users make comments, answer questions, post a question/topic, upvote/downvote other's comments/post.
UC-9	Moderate Posts	Administrator needs to maintain control of the system. Therefore, if any post or comment is reported as inappropriate or irrelevant, administrator has privilege to remove that post and if a user's post/comment is reported more than 3 times, system administrator will have to block that account temporarily to avoid any more suspicious activities.

UC-10	Upvote/Downvote	User presses the triangle or inverse triangle next to the post or comments in order to upvote/downvote it. System automatically adds points to the user rating if their post/comments get upvoted and subtract the point if otherwise.
UC-11	Post in Opportunity Pages	Only premium users can create post in this page. User will be asked to provide a heading, a short description a long description, category tags, and optionally an image describing the opportunity. Then it will wait for the administrator/moderator to approve of the post before it's accessible to the public.
UC-12	View Opportunity Page	Opportunity page includes posts about scholarships, university programs, job announcements, internships, competitions, workshops, courses, fairs, and volunteer opportunities. Users can view posts in the page, and they can filter it by tags or category.
UC-13	Bookmark Posts	Once they see any post they're interested in, users can bookmark it for later view. To access the bookmarked posts, user clicks on the bookmark icon.
UC-14	Follow Other Users	Users click on subscribe/follow other users to get notification about their posts. And they click on unfollow to stop getting the notification.
UC-15	Create Private Chat	Users reach out and connect to others through chat function, thus it increases the amount of interaction between users. They will first need to go to the other user's profile page and click on the message icon in order to start chat. Alternatively, user can go to their contact book in order to search for the person they want to start chat with.
UC-16	Search	Users search for existing posts, topics, questions through the search bar by inputting keywords.
UC-17	Filter	Each post, forum question, and topic has tags associated to them to identify the category. Users select categories of their interest in order to filter posts.
UC-18	Access Email System	System enables users to connect with each other through other communication channels such as email as well.
UC-19	Access Exclusive Pages	Only users with predetermined ratings will have access to exclusive pages.
UC-20	Get Notification	Users get notifications for any account related activities or messages.

UC-21	Manage User Account	Admins view, add, update, and delete any user account and the user profile information including profile photo, personal information, educational background, work experiences, volunteer experiences.
-------	---------------------	--

#### 4.3.b. Use Case Diagram

1. Define Actors explained in diagram:

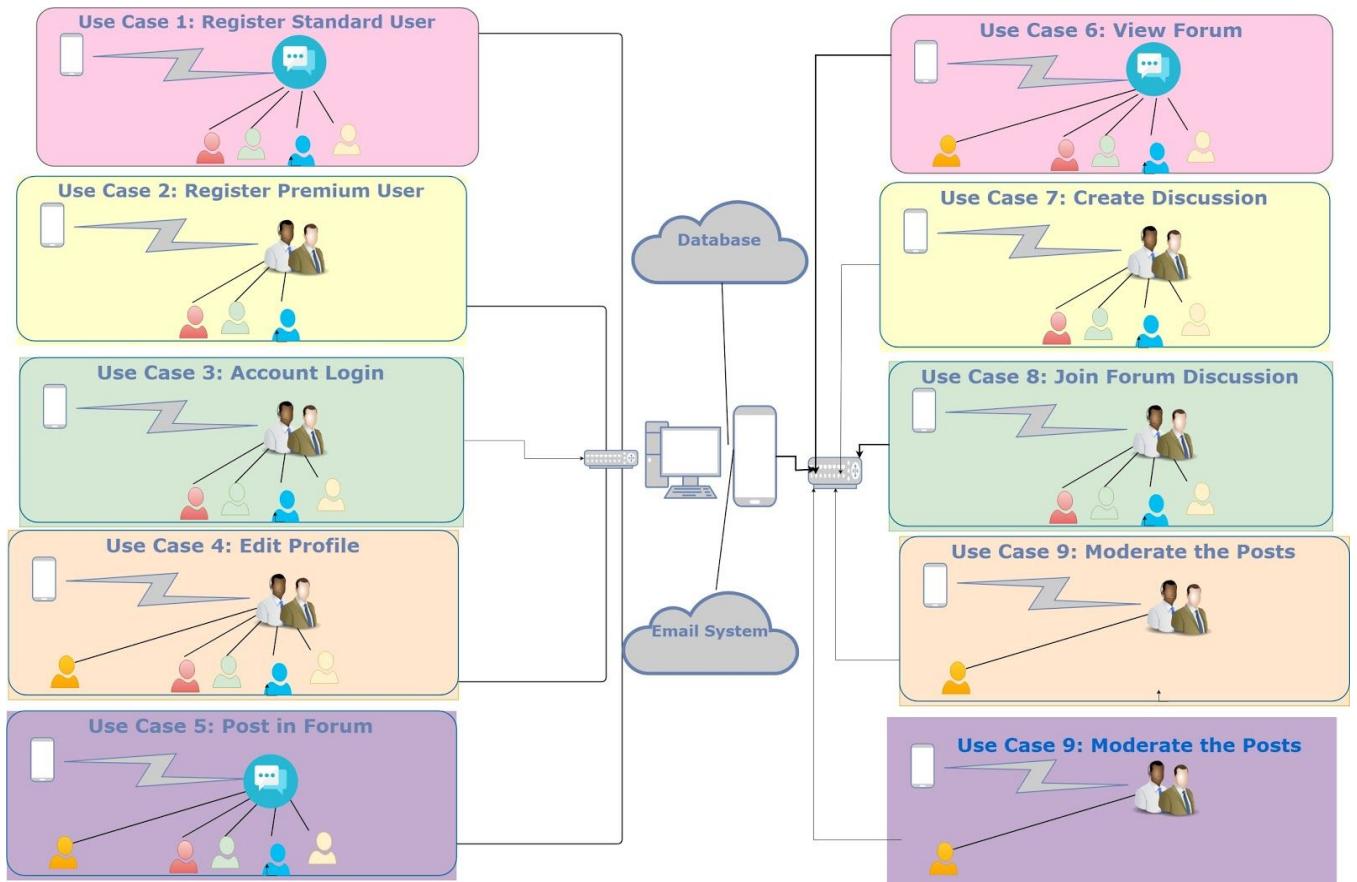
- **Student:** target user
- **Job seeker:** target user
- **Recruiter:** career vacancy open and recruiter
- **Admin:** system administrator
- **Database:** store input data
- **Email System:** communication and network tool
- **Teacher:** school representative, scholars, professors
- **Tutor:** mentors for the mentorship program

1) Diagram to show interactions between the actors and use cases 1-10

a) Explain diagram in plain table

UC-1	Register Standard User	Students, job seekers, tutor, teacher, recruiter
UC-2	Register Premium User	Teacher, tutor, recruiter
UC-3	Account Login	Students, job seekers, tutor, teacher, recruiter
UC-4	Edit Profile	Students, job seekers, tutor, teacher, recruiter
UC-5	Post in Forum	Students, job seekers, tutor, teacher, recruiter, administrator
UC-6	View Forum	Students, job seekers, tutor, teacher, recruiter, administrator
UC-7	Create Discussion	Students, job seekers, tutor, teacher, recruiter
UC-8	Join Forum Discussion	Students, job seekers, tutor, teacher, recruiter
UC-9	Moderate the Post	Administrator
UC-10	Upvote/Downvote	Administrator

b. Diagram: Actor and Use Cases Interaction



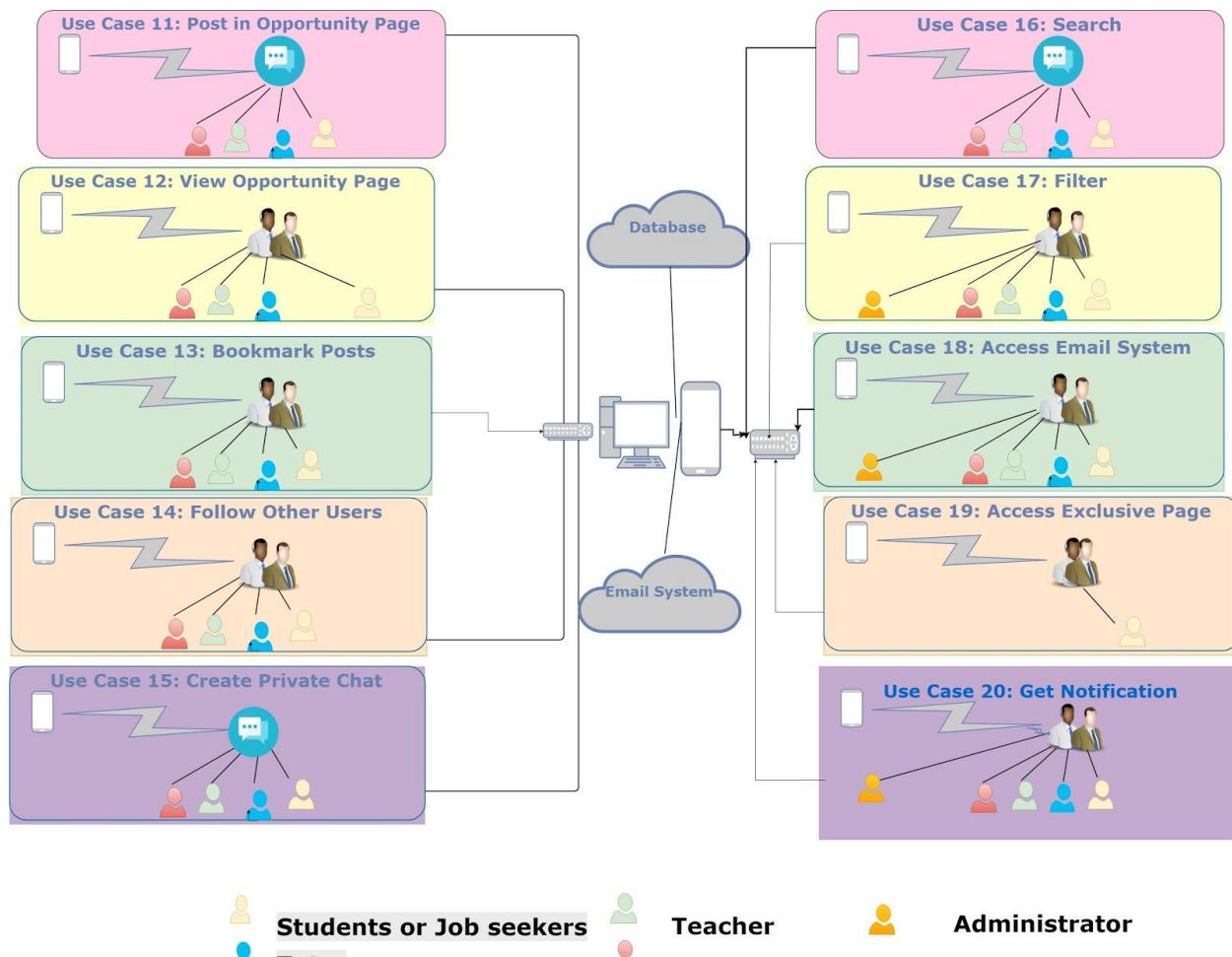
2) Diagram to show interactions between the actors and use cases 10-20

a) Explain diagram in plain table

UC-11	Post in opportunity page	tutor, teacher, recruiter
UC-12	View opportunity page	Students, job seekers, tutor, teacher, recruiter
UC-13	Bookmark Posts	Students, job seekers, tutor, teacher, recruiter
UC-14	Follow other users	Students, job seekers, tutor, teacher, recruiter
UC-15	Create Private Chat	Students, job seekers, tutor, teacher, recruiter
UC-16	Search	Students, job seekers, tutor, teacher, recruiter

UC-17	Filter	Students, job seekers, tutor, teacher, recruiter
UC-18	Access Email System	Students, job seekers, tutor, teacher, recruiter
UC-19	Access Exclusive Page	Students, job seekers
UC-20	Get Notification	Students, job seekers, tutor, teacher, recruiter
UC-21	User Profile	Students, job seekers, tutor, teacher, recruiter
UC-22	Manage User Account	Administrator

b. Diagram: Actors and Use cases Interaction



4.3.c. Traceability Matrix

	Priority	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-1

	Weight											0
REQ-1	5	X										
REQ-2	5		X									
REQ-3	4	X	X									
REQ-4	3	X	X	X	X							
REQ-5	5											X
REQ-6	5					X	X					X
REQ-7	5						X					
REQ-8	3											
REQ-9	4					X		X				
REQ-10	4					X	X	X	X	X		
REQ-11	3											
REQ-12	3											
REQ-13	1	X	X	X	X	X	X	X	X	X	X	
REQ-14	5	X	X	X	X	X	X	X	X	X	X	
REQ-15	2											
REQ-16	2											
REQ-17	2			X								
REQ-18	3			X								
REQ-19	5											X
REQ-20	4											
REQ-21	4	X	X	X	X	X	X	X	X	X	X	
REQ-22	3	X	X	X	X	X	X	X	X	X	X	
REQ-23	4	X	X	X	X	X	X	X	X	X	X	
REQ-24	2	X	X	X								
REQ-25	3											

REQ-26	4	X	X	X	X	X	X	X	X	X	X	X
REQ-27	5	X	X	X	X	X	X	X	X	X	X	X
REQ-28	3	X	X	X	X	X	X	X	X	X	X	X
REQ-29	4	X	X	X	X	X	X	X	X	X	X	X
REQ-30	4	X	X	X	X	X	X	X	X	X	X	X
REQ-31	4	X	X	X	X	X	X	X	X	X	X	X
REQ-32	5	X	X	X	X							
REQ-33	4											
Total PW		64	64	57	41	54	55	49	45	45	45	47

	Priority Weight	UC-11	UC-12	UC-13	UC-14	UC-15	UC-16	UC-17	UC-18	UC-19	UC-20	UC-21
REQ-1	5											
REQ-2	5											
REQ-3	4											
REQ-4	3											
REQ-5	5											
REQ-6	5											
REQ-7	5		X									
REQ-8	3			X								
REQ-9	4	X										
REQ-10	4	X	X									
REQ-11	3	X	X							X		
REQ-12	3									X		
REQ-13	1	X	X	X								

REQ-14	5	X	X	X	X	X							
REQ-15	2					X			X				
REQ-16	2				X				X		X		
REQ-17	2												
REQ-18	3										X		
REQ-19	5												
REQ-20	4						X	X					
REQ-21	4	X	X	X	X	X	X	X	X	X			
REQ-22	3	X	X	X	X	X	X	X	X	X			
REQ-23	4	X	X	X	X	X	X	X	X	X	X		
REQ-24	2												
REQ-25	3						X	X					
REQ-26	4	X	X	X	X	X	X	X	X	X	X		
REQ-27	5	X	X	X	X	X	X	X	X	X	X		
REQ-28	3	X	X	X	X	X	X	X	X	X	X		
REQ-29	4	X	X	X	X	X	X	X	X	X	X		
REQ-30	4	X	X	X	X	X	X	X	X	X	X		
REQ-31	4	X	X	X	X	X	X	X	X	X	X		
REQ-32	5				X		X						
REQ-33	4												X
Total PW		52	53	44	47	42	47	42	39	41	33	28	

#### 4.3.d. Fully-Dressed Description

The following use cases are important to the user requirement specification, therefore they're selected for a fully-dressed description:

- Register Standard User
- Register Premium User
- Account Login

- Post In Forum
- View Forum
- Post in Opportunity Pages
- Edit User Profile
- Manage User Account

### **UC-1 : Register Standard User**

**Related Requirements:** REQ-1, 3,4,13,14,21,22,23,24,26,27,28,29,30,31

**Initiating Actor:** Standard User (Student, Job Seeker)

**Actor's Goal:** To create a standard account with nEdCom.

**Participating Actor:** Database, Email System

**Preconditions:** User's email is not associated with any nEdCom account yet. User need to provide first name, last name, valid email address, username, and password.

**Postconditions:** Upon registration, a verification email has been sent to user notifying of account confirmation. User has clicked on the confirmation link in the email, user account is successfully created, and the system has notified user to add additional information to the profile including: personal information, education background, careers, interests, and projects.

#### **Main Success Scenario:**

- 1. **User** select on “Register As Standard User”
- ← 2. **System** prompts student/job seeker to enter credentials
- 3. **User** enter their credentials
- 4. **System** verifies that username and password meet requirement
- 5. **System** checks from the **database** if email and username is already in use
- ← 6. **System** verifies email and username
- 7. **System** sends confirmation code to user’s email using the **email system**.
- ← 8. **System** notifies **user** to check email and prompt user to enter the confirmation code
- 9. **User** enters confirmation code
- 10. **System** checks if confirmation code is correct
- ← 11. **System** verifies confirmation code
- 12. **System** activates user account
- ← 13. **System** notifies **user** of successful registration
- ← 14. **System** asks **user** to enter profile information
- 15. **User** enters profile information
- 16. **Database** stores user’s profile information

#### **Extensions (Alternative Scenarios):**

- 3a. **User** enters invalid password
  - ← 1. **System** prompts user to enter another password
- 3b. **User** enter email or username that has already been used.
  - ← 1. **System** prompts user to enter another email or username
- 9a. **User** enters incorrect confirmation code

← 1. **System** prompts user to re-enter confirmation code

## UC-2 : Register Premium User

**Related Requirements:** REQ-2, 3,4,13,14,21,22,23,24,26,27,28,29,30,31

**Initiating Actor:** User (Faculty, Recruiter, Advisor, Event Organizer)

**Actor's Goal:** To create a premium account with nEdCom

**Participating Actor:** Administrator, Database, Email System

**Preconditions:** User doesn't have a nEdCom account yet. They are required to provide first name, last name, valid company or university email address, phone number, photo of national ID, and proof of employment.

**Postconditions:** Administrator has verified all the above information and approved of creation of the account. User has clicked on the confirmation link in the email, user account is successfully created with premium type, and the system has notified user to add additional information to the profile including: personal information, education background, careers, interests, and projects.

- 1. **User** select on “Register As Premium User”
- ← 2. **System** prompts student/job seeker to enter credentials
- 3. **User** enter their credentials
- 4. **System** verifies that username and password meet requirement
- 5. **System** checks from the **database** if email and username is already in use
- ← 6. **System** verifies email and username
- 7. **System** sends confirmation code to user's email using the **email system**.
- ← 8. **System** notifies **user** to check email and prompt user to enter the confirmation code
- 9. **User** enters confirmation code
- 10. **System** checks if confirmation code is correct
- ← 11. **System** verifies confirmation code
- 12. **System** alerts administrators to review user's registration request
- ← 13. **System** notifies user to wait for administrator approval
- ← 14. **Administrator** approves of user's registration request
- 15. **System** activates user account
- ← 16. **System** notifies **user** of successful registration
- ← 17. **System** asks **user** to enter profile information
- 18. **User** enters profile information
- 19. **Database** stores user's profile information

### Extensions (Alternative Scenarios):

3a. **User** enters invalid password

← 1. **System** prompts user to enter another password

3b. **User** enter email or username that has already been used.

← 1. **System** prompts user to enter another email or username  
9a. **User** enters incorrect confirmation code  
← 1. **System** prompts user to re-enter confirmation code

### UC-3 : Account Login

**Related Requirements:** REQ-4,13,14,17,18, 21,22,23,24,26,27,28,29,30,31

**Initiating Actor:** Administrator, Student, Job Seeker, Faculty, Recruiter, Advisor, Event Organizer

**Actor's Goal:** To login to access features of the system.

**Participating Actor:** Database

**Preconditions:** User already has an account. The login page requires user email and password.

**Postconditions:** The credentials match. User is logged in.

#### Main Success Scenario:

→ 1. **User** select on Login  
← 2. **System** prompts **user** to enter credentials  
→ 3. **User** enter credentials  
→ 4. **System** checks user's credentials  
← 5. **System** verifies user's credentials  
← 6. **System** grants user access

#### Extensions (Alternative Scenarios):

3a. User enter wrong credentials  
← 1. System prompts user to re-enter credentials

### UC-5 : Post In Forum

**Related Requirements:** REQ-6,9,10,13,14, 21,22,23,26,27,28,29,30,31

**Initiating Actor:** Student, Job Seeker, Administrator, Faculty, Recruiter, Advisor, Event Organizer

**Actor's Goal:** To post questions or start discussion topics.

**Participating Actor:** Database

**Preconditions:** User of any account type can create a post in forum.

**Postconditions:** The forum is saved to the system database

#### Main Success Scenario:

→1. **User** enter the forum page  
←2. **User** select function "create post"  
→3. **System** checks if user is eligible to post

- ←4. **System** verifies user's eligibility
- ←5. **System** prompts user to fill in information about the post
- 6. **User** submit information
- ←7. **System** verifies if information meets the requirement
- 8. **System** add the post to the database

**Extensions (Alternative Scenarios):**

- 6a. **User** submit information that don't meet requirement
- ←1. **System** prompts user to edit and resubmit the post.
- 3a. **User** doesn't have eligibility to post
- ←1. **System** alert to user of the ineligibility

### UC-6 : View Forum

**Related Requirements:** REQ-6,7,10,13,14, 21,22,23,26,27,28,29,30,31

**Initiating Actor:** User (Administrator, Student, Job Seeker, Faculty, Recruiter, Advisor, Event Organizer)

**Actor's Goal:** To access questions / comments / discussion topics.

**Participating Actor:** Database

**Preconditions:** User of any account type can browse discussion topics in the forum.

**Postconditions:** System has retrieved data from the database and displayed to the webpage.

**Main Success Scenario:**

- 1. **User** enter the forum page
- 2. **System** asks the database to list all the posts for user to view
- ←3. **Database** returns list of posts
- ←4. **System** displays all posts to the webpage

### UC-11 : Post In Opportunity Pages

**Related Requirements:** REQ-9,10,11,13,14, 21,22,23,26,27,28,29,30,31

**Initiating Actor:** Premium User (Faculty, Recruiter, Advisor, Event Organizer)

**Actor's Goal:** To post information related to job announcement, scholarships, internship, networking events, and other opportunities.

**Participating Actor:** Standard User (Student, Job Seeker), Administrator, Database

**Preconditions:** User must have a premium account type in order to create post in opportunity page. User will need to provide the headline, short description, long description, category tags, and optionally an image to describe the opportunity.

**Postconditions:** Administrator / moderator has reviewed and approved of the post. It gets saved to system database and the post becomes accessible by public.

**Main Success Scenario:**

- 1. **User** enter the opportunity page
- 2. **User** select on “create post”
- 3. **System** checks in the **database** if the user is eligible to post
- ←4. **System** verifies user eligibility
- 5. **System** checks in the **database** to list all tags/categories
- ←6. **Database** return list of all tags/categories
- ←7. **System** display form for user to input post information, including the tags that user need to select to associate with the post
- 8. **User** submit the information
- ←9. **System** verifies that the information meet the requirements
- 10. **System** alerts **administrator** to review the post
- ←11. **System** notifies user to wait for administrator approval
- ←12. **Administrator** approves of the post
- 13. **System** stores the post to the **database**
- ←14. **System** notifies user of the approval.

**Extensions (Alternative Scenarios):**

- 8a. **User** fill in information that doesn't meet requirement
- ←1. **System** prompts user to edit and resubmit the post
- 2. **User** edits and resubmits the post
- 12a. **Administrator** rejects the post
- ←14. **System** notifies user of the reject

**UC-4 : Edit Profile**

**Related Requirements:** REQ-4,13,14, 21,22,23,26,27,28,29,30,31

**Initiating Actor:** Administrator, Student, Job Seeker, Faculty, Recruiter, Advisor, Event Organizer

**Actor's Goal:** To edit profile details such as employment history, educational background, volunteering activities.

**Participating Actor:** Database

**Preconditions:** User already is logged in to the profile page. User may edit fields in the profile page.

**Postconditions:** User information is updated and saved to the database.

**Main Success Scenario:**

- 1. **User** enters profile page
- 2. **System** fetches user profile record

← 3. **Database** returns user profile record  
→ 3. **User** selects function Edit Profile  
← 4. **System** verify if the information entered meets the requirement  
→ 5. **System** saves the result  
← 6. **System** displays the result to user showing that information is updated successfully

**Extensions (Alternative Scenarios):**

4a. User enter invalid input  
← 1. System prompts user to re-enter information

### UC-21 : Manage User Account

**Related Requirements:** REQ-26,27,28,29,30,31,33

**Initiating Actor:** Administrator

**Actor's Goal:** To manage account details and permissions of all user accounts

**Participating Actor:** Database

**Preconditions:** User is logged in as administrator. User need to select a particular user and update their profile.

**Postconditions:** The user profile is updated successfully.

**Main Success Scenario:**

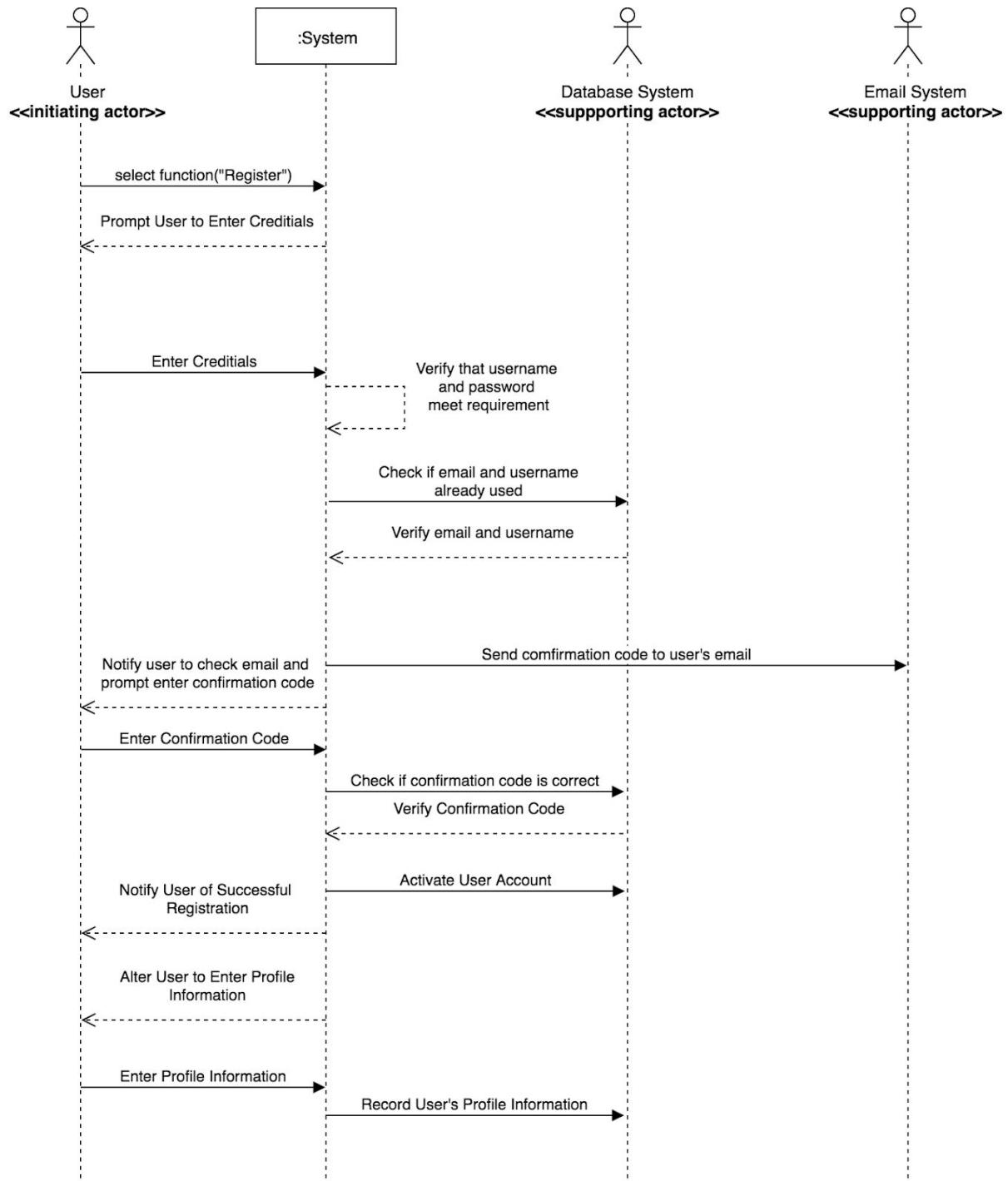
→ 1. **Administrator User** select on function “manage user”  
→ 2. **System** fetches user records  
← 3. **Database** returns user records  
→ 4. **Administrator User** selects a particular user account  
→ 5. **System** fetches user’s profile record  
← 6. **Database** returns user’s profile record  
→ 7. **Administrator User** selects function “Update User Account”  
← 8. **System** verify if the information entered meets the requirement  
→ 9. **System** saves the result  
← 10. **System** displays the result to user showing that information is updated successfully

**Extensions (Alternative Scenarios):**

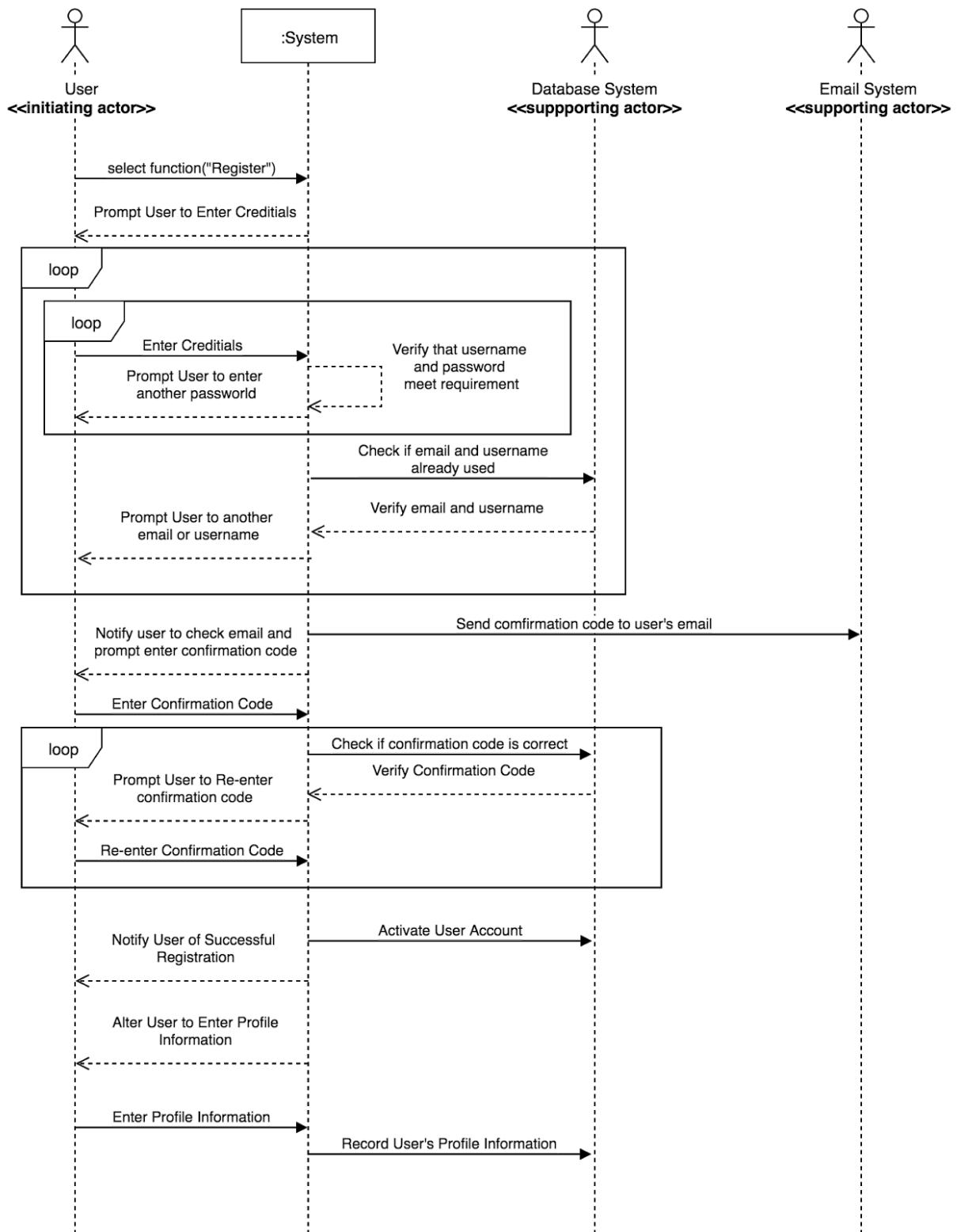
8a. User enter invalid input  
← 1. System prompts user to re-enter information

#### 4.4. System Sequence Diagrams

##### Use Case 1: Registering Standard User Main Success Scenario

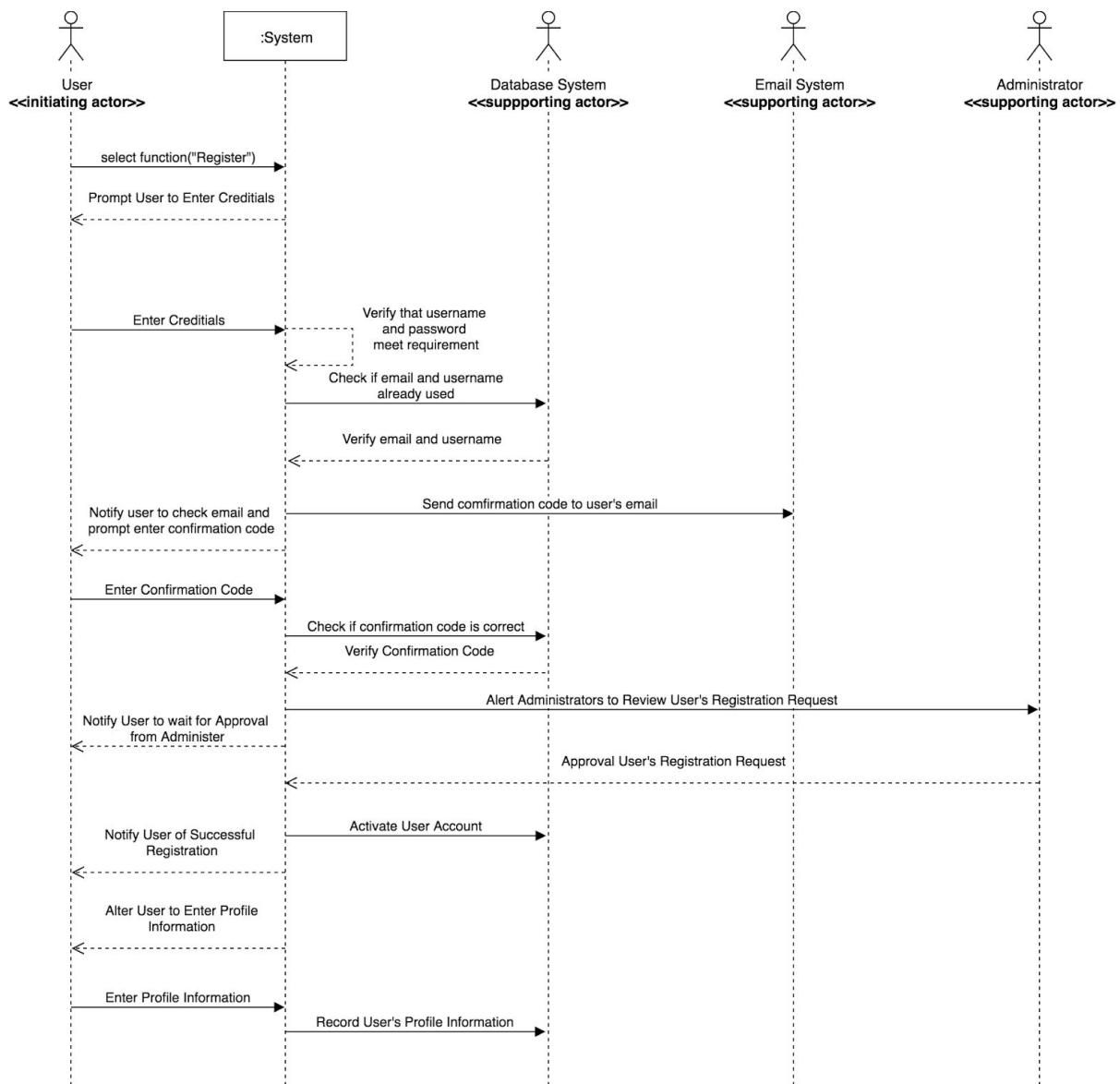


## Use Case 1: Registering Standard User Alternative Scenario

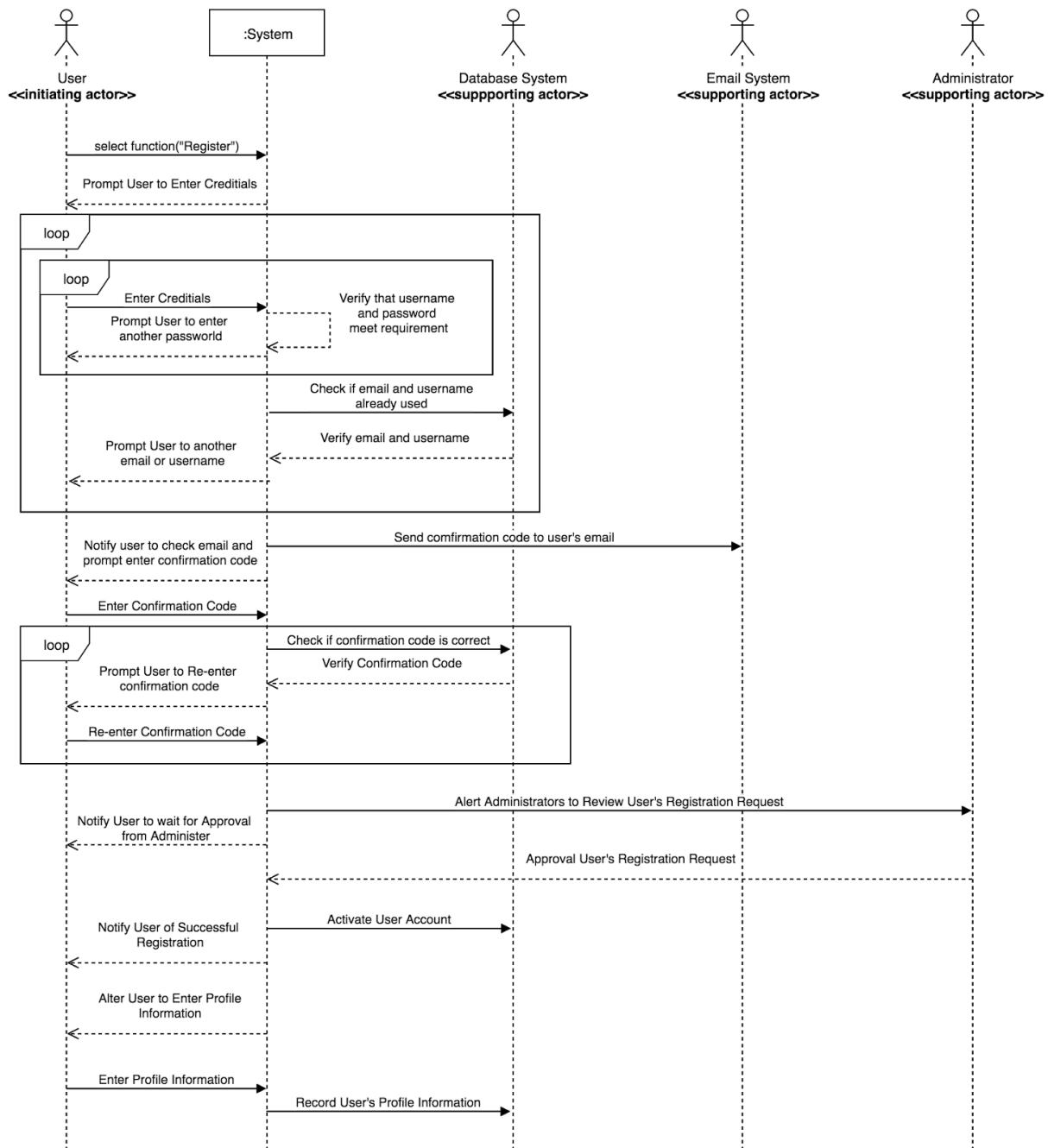


## Use Case 2: Registering Premium User

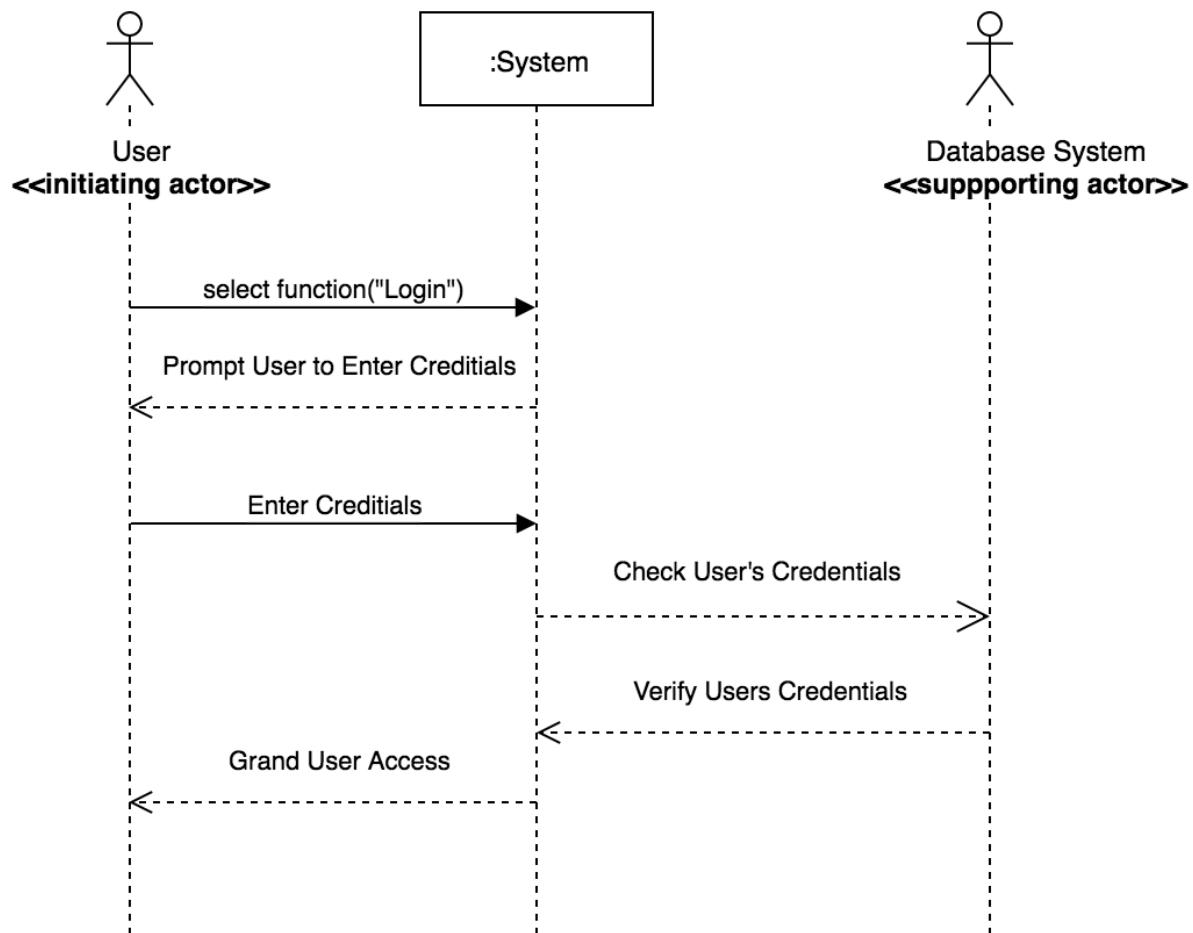
### Main Success Scenario



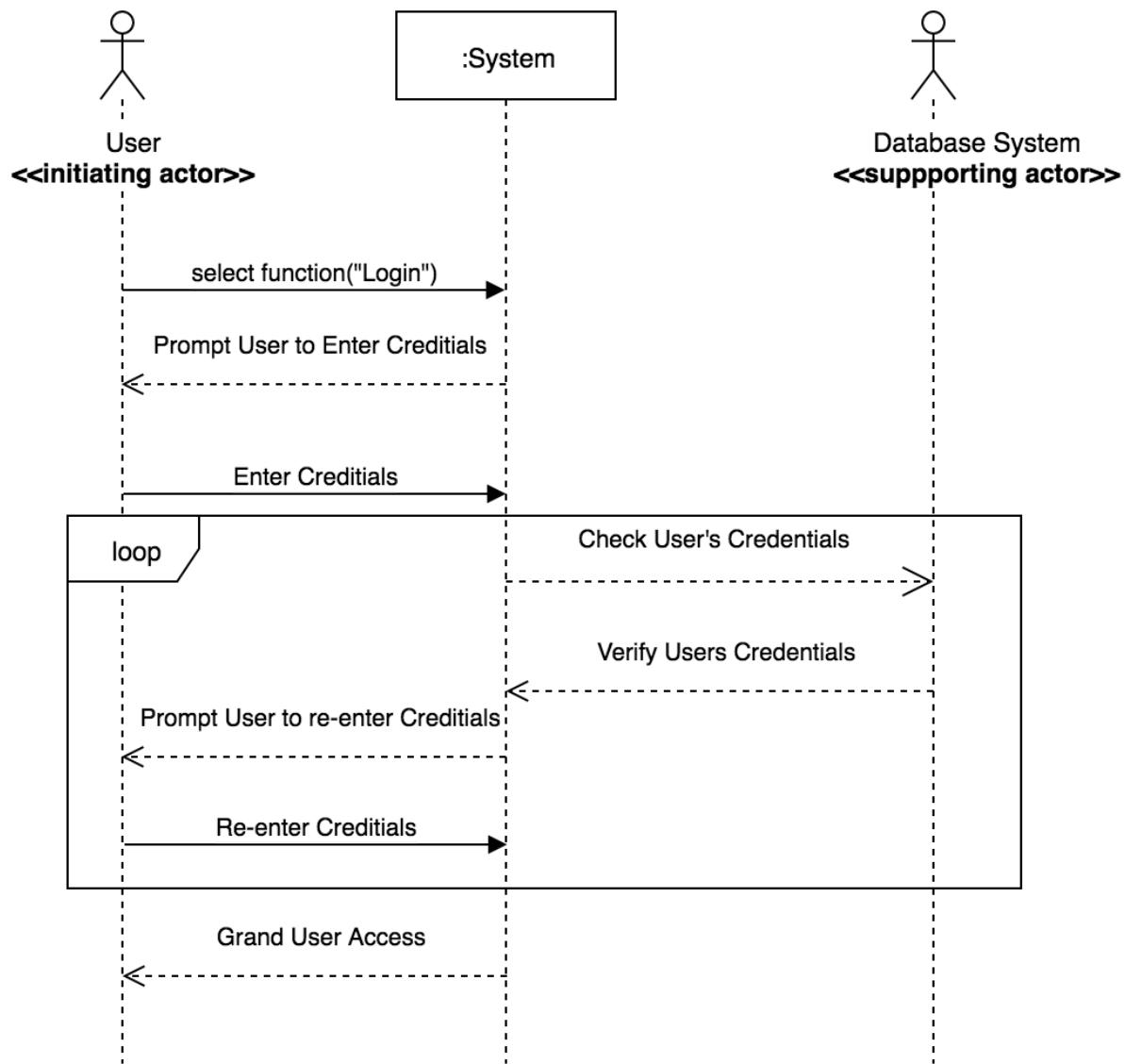
## Use Case 2: Registering Premium User Alternative Scenario



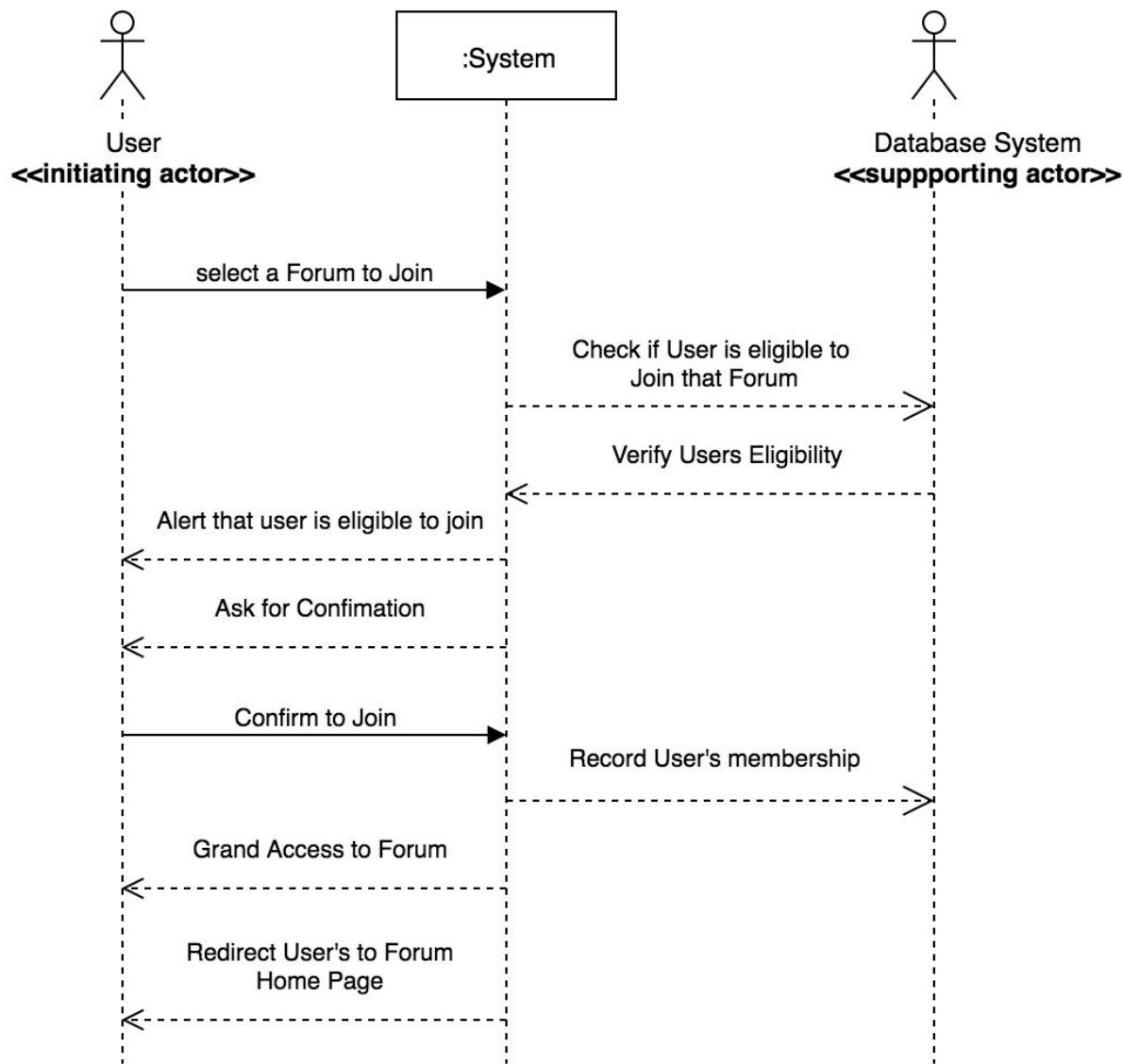
**Use Case 3: Login**  
**Main Success Scenario**



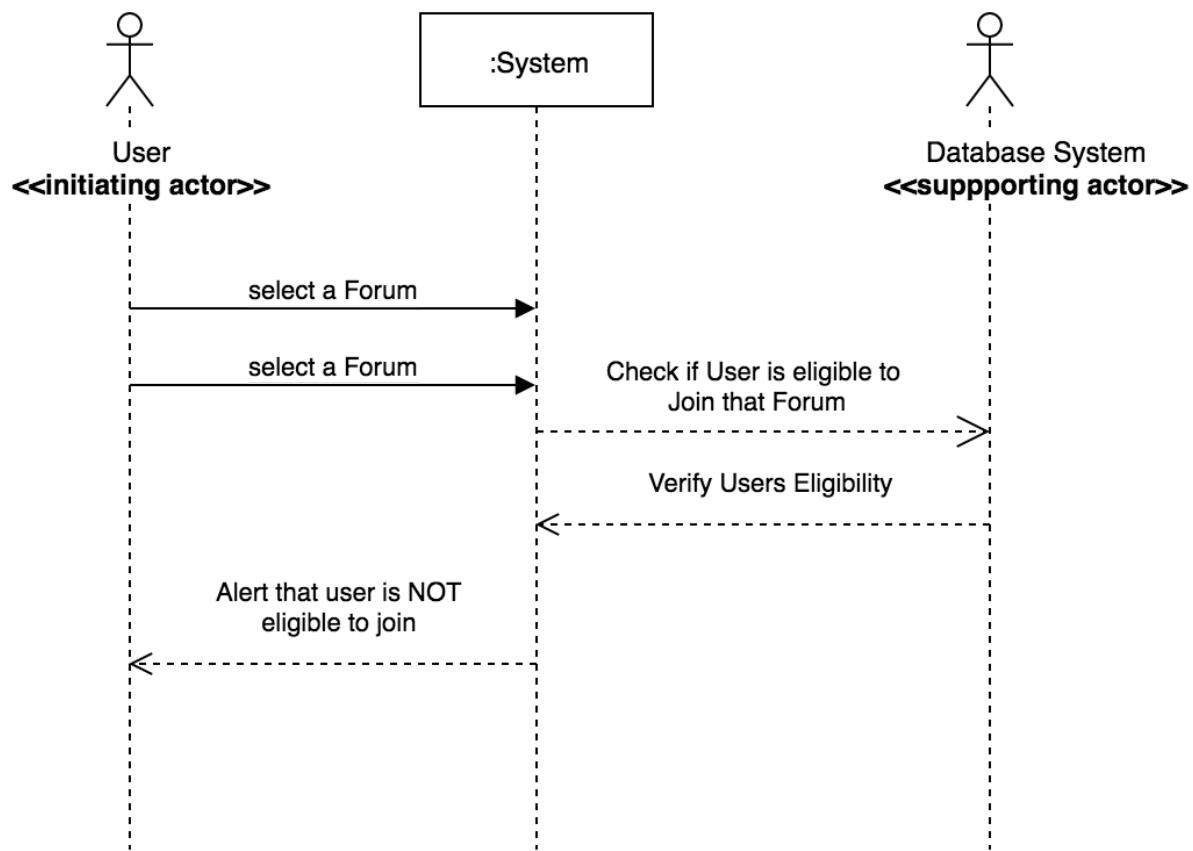
**Use Case 3: Login**  
**Alternative Scenario**



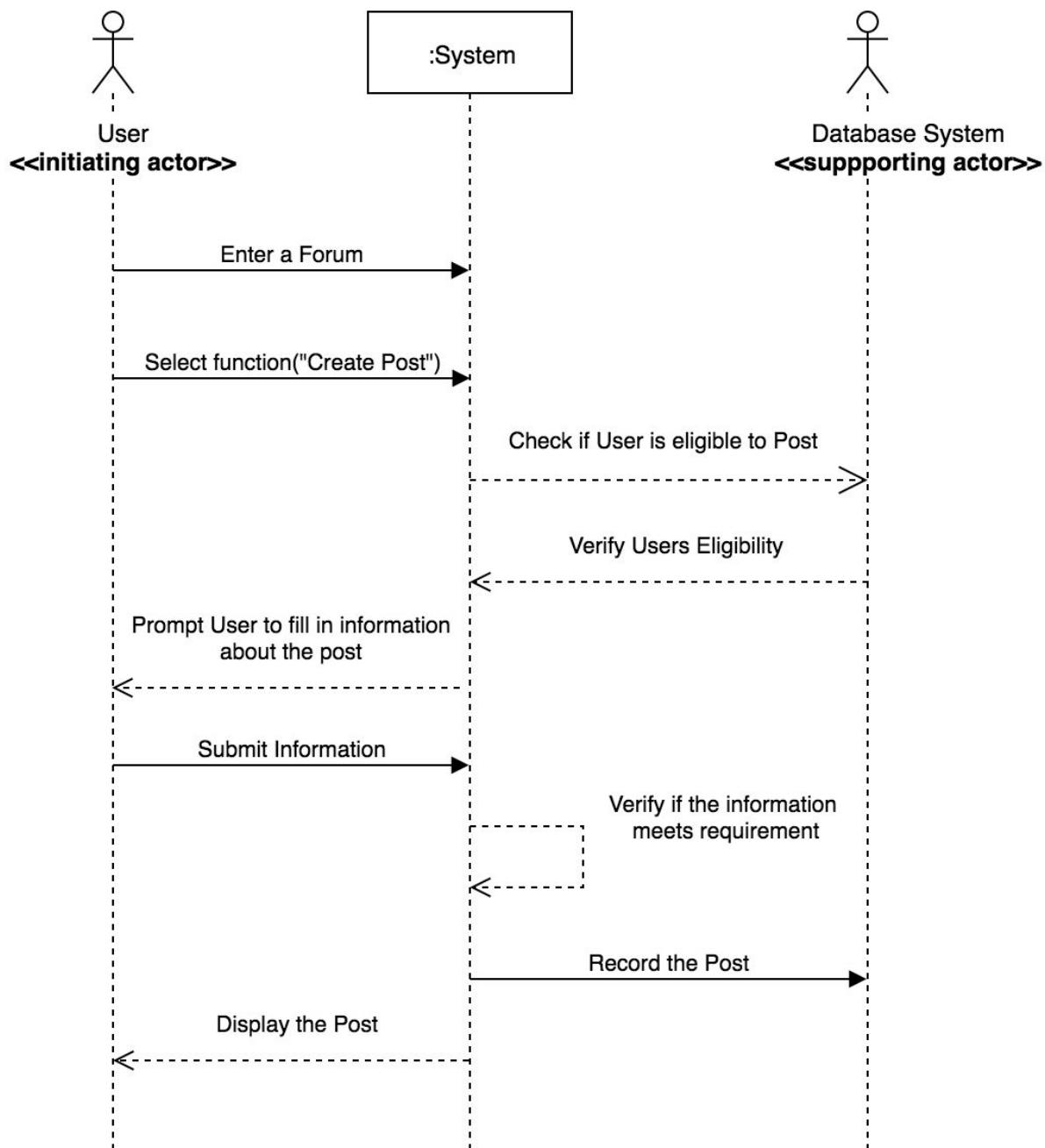
**Use Case 6: Join Forum**  
**Main Success Scenario**



**Use Case 6: Join Forum**  
**Alternative Scenario**

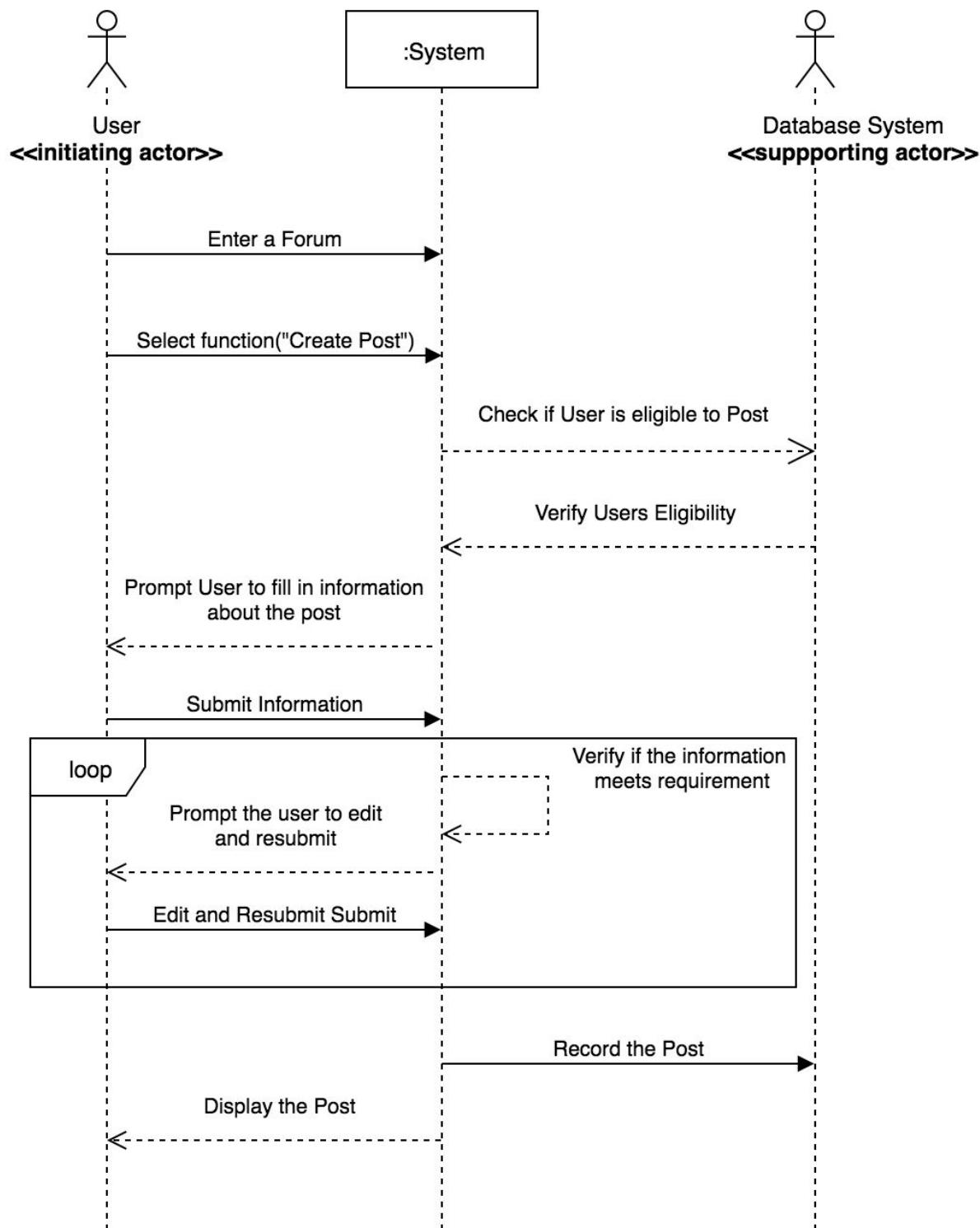


**Use Case 5: Post in Forum**  
**Main Success Scenario**

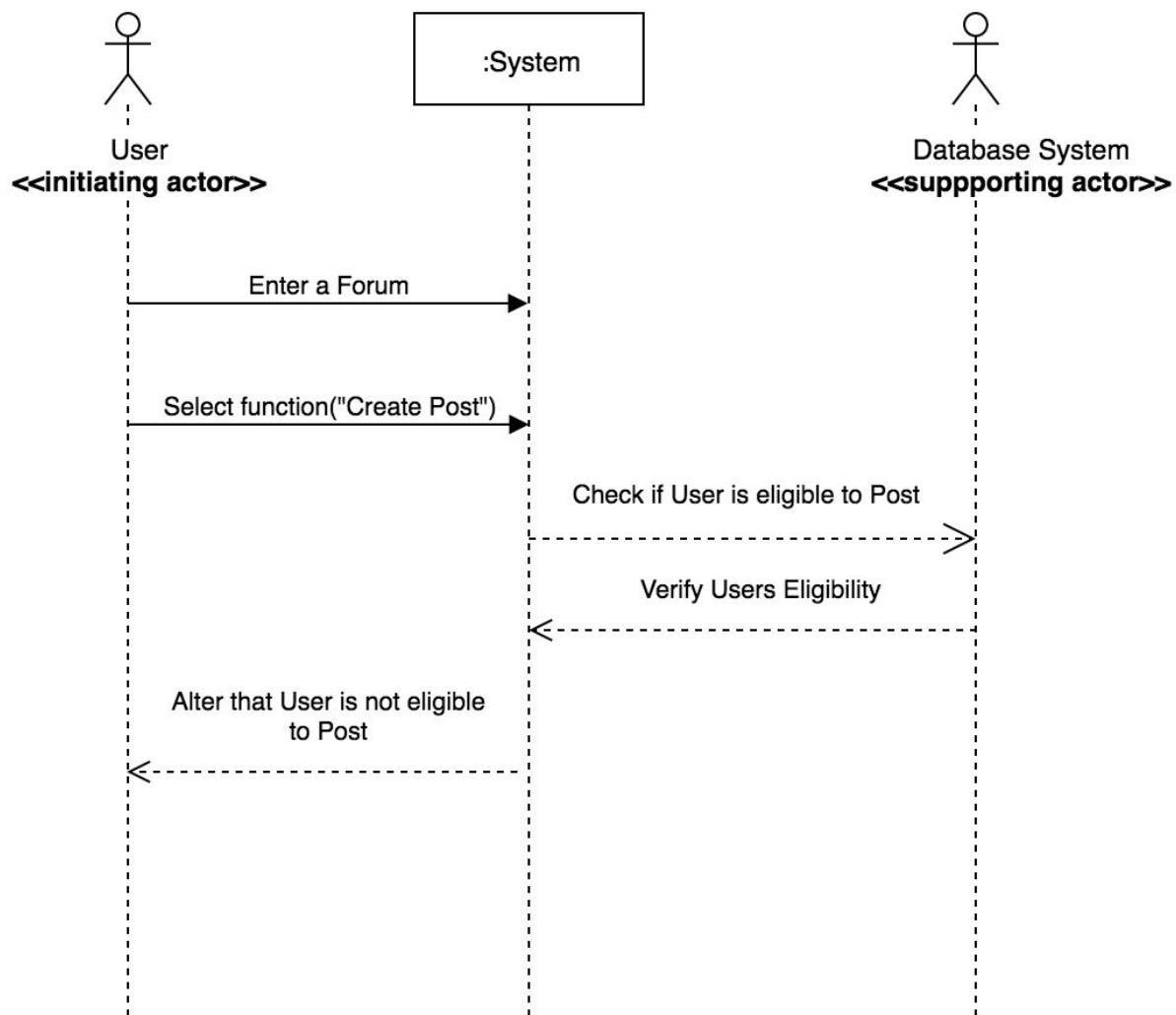


## Use Case 5: Post in Forum

### Alternative Scenario 1

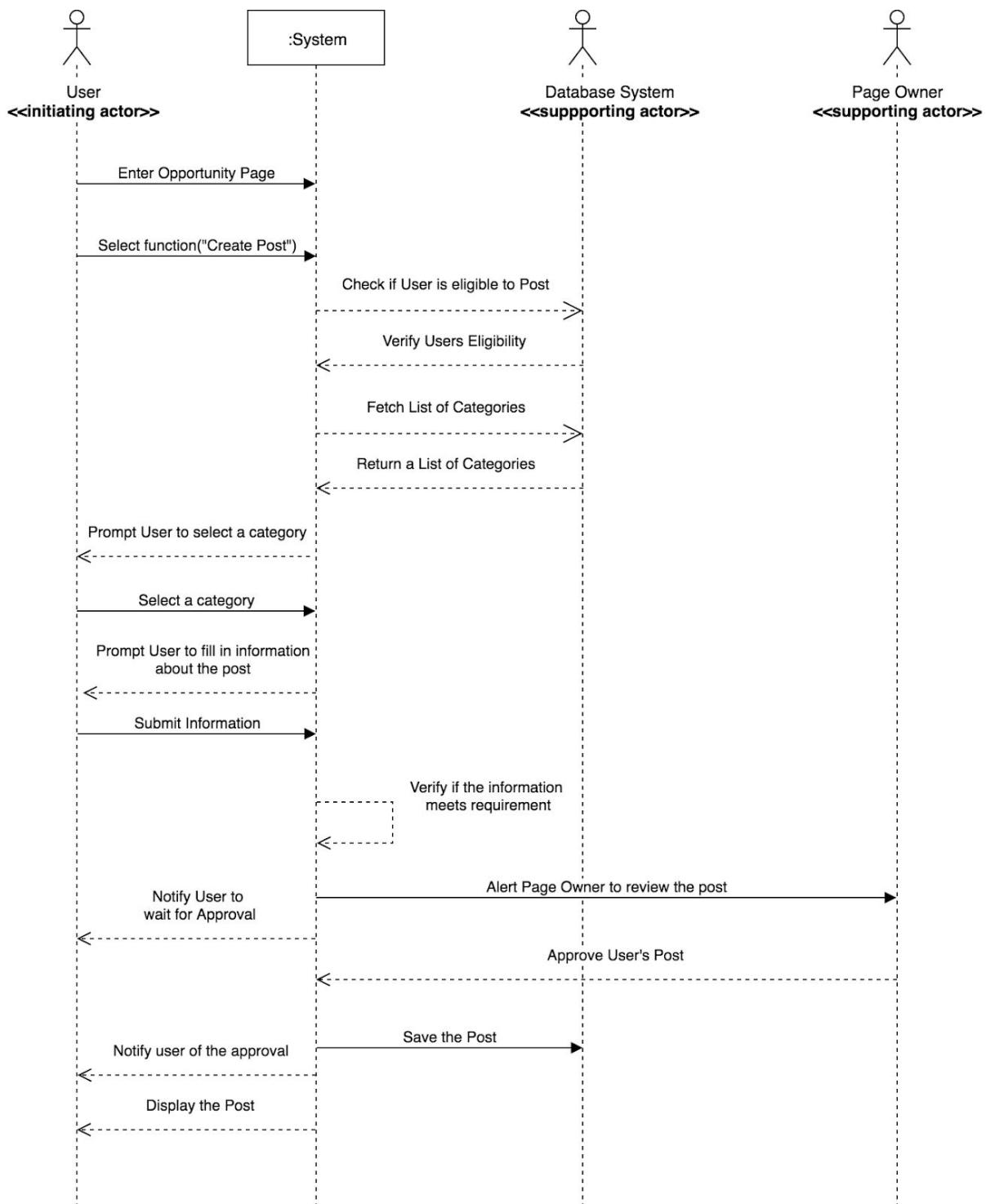


**Use Case 5: Post in Forum**  
**Alternative Scenario 2**



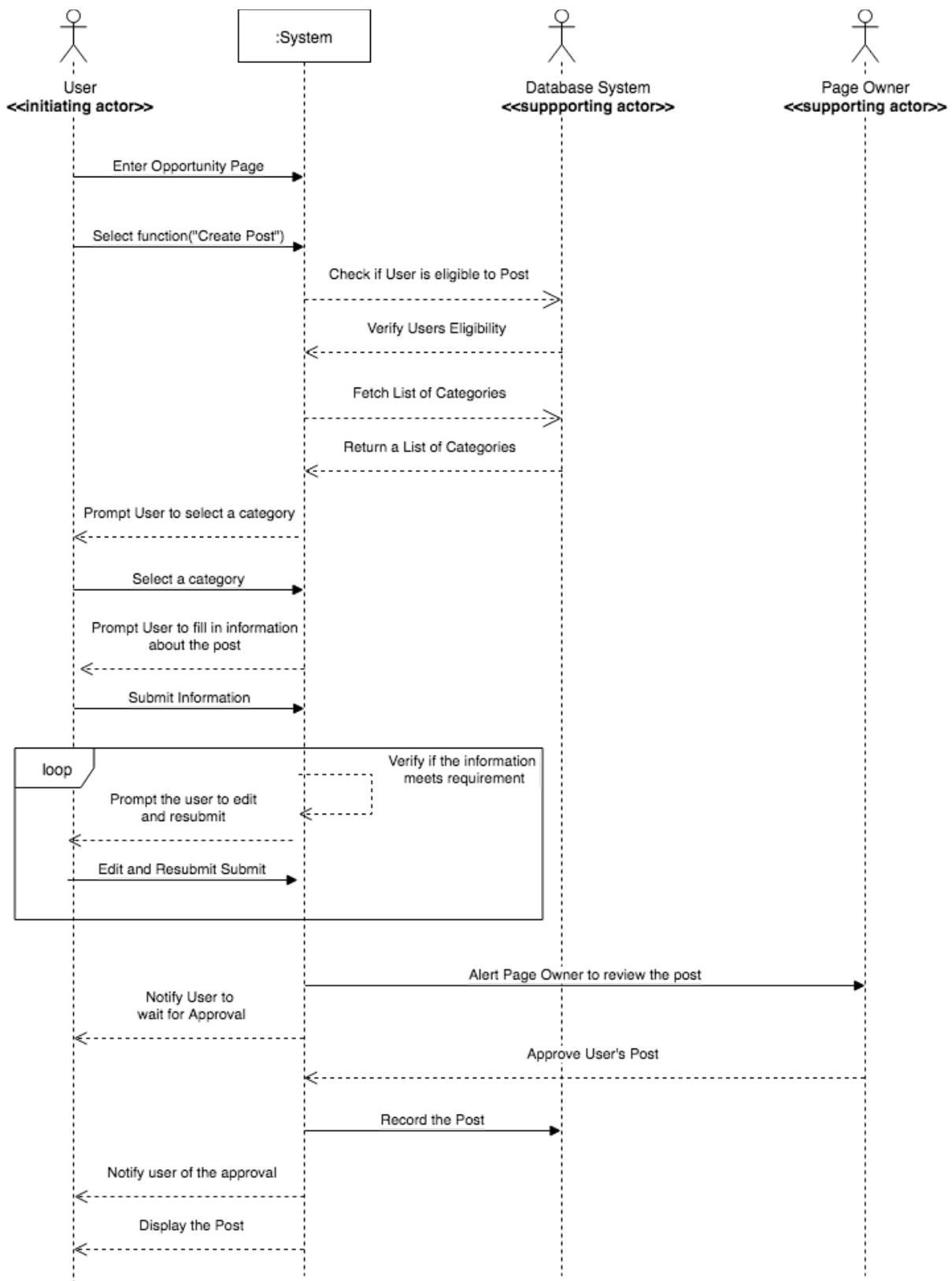
## Use Case 11: Create Post in Opportunity Page

### Main Success Scenario



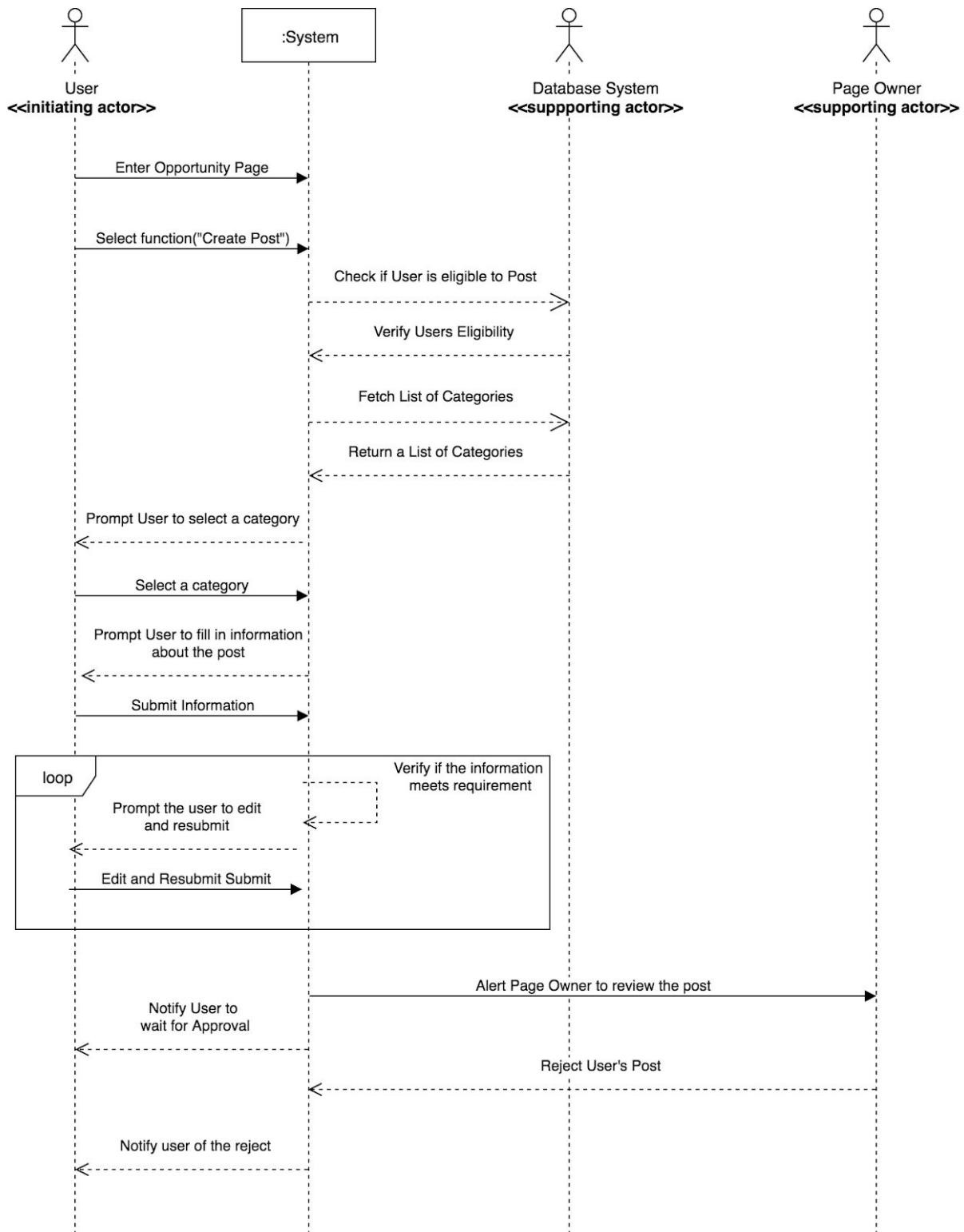
## Use Case 11: Create Post in Opportunity Page

### Alternative Scenario 1

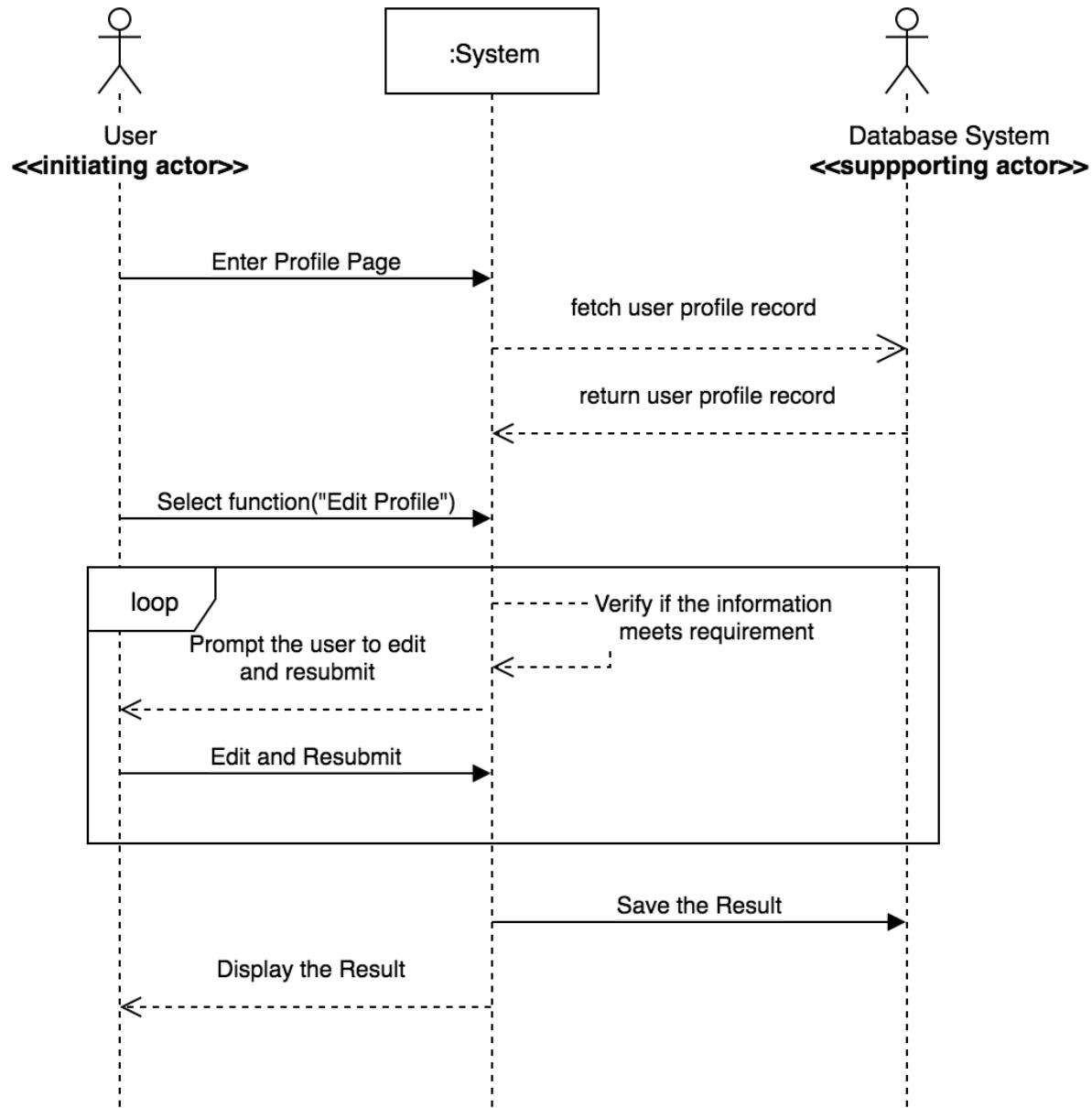


## Use Case 11: Create Post in Opportunity Page

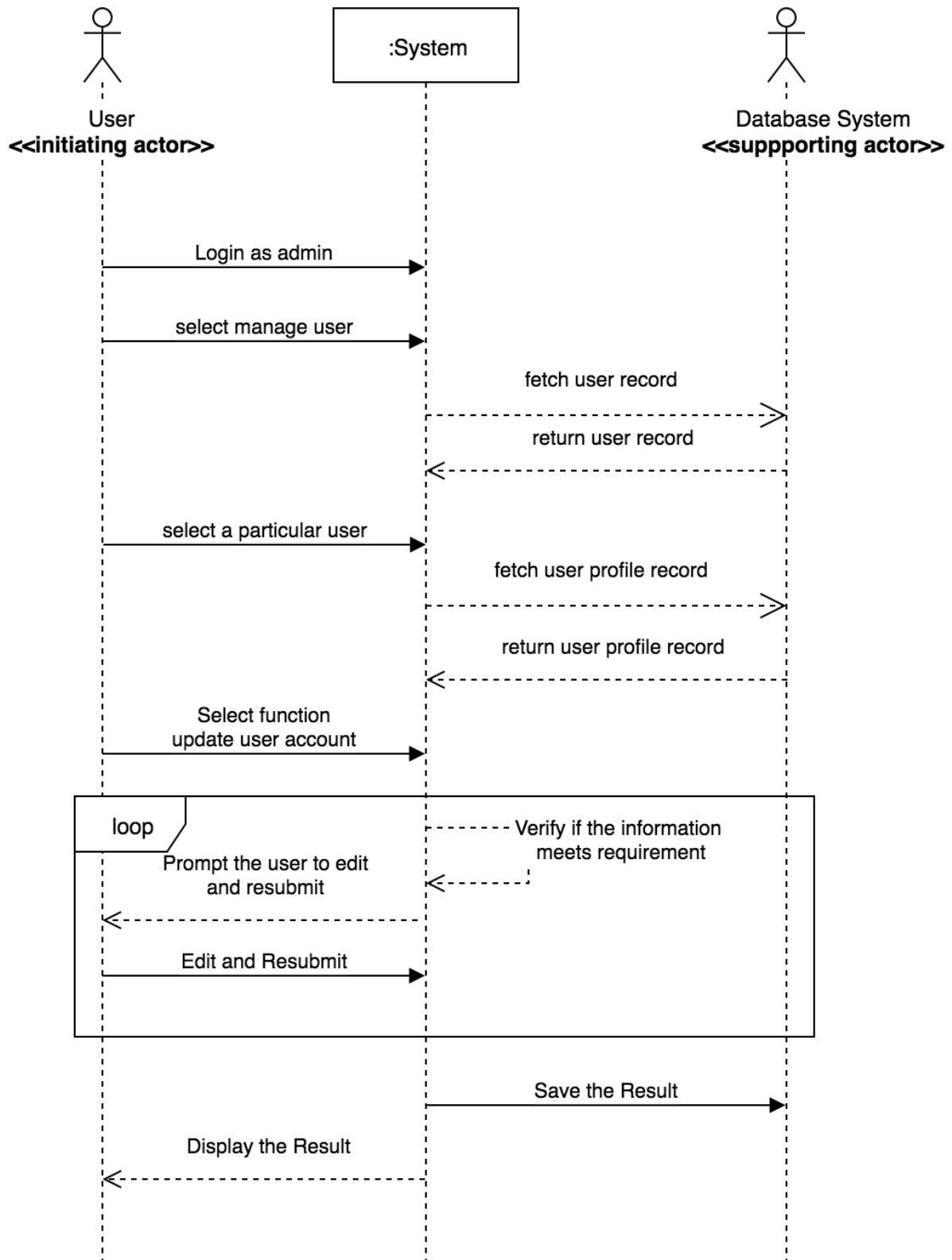
### Alternative Scenario 2



### Use Case 4: Edit Profile Page



### Use Case 21: Manage User Account (Update User Account)



#### 4.5. User Effort Estimation

<b>Use Case</b>	<b>Action</b>	<b>Description</b>	<b>Total Click/KeyStrokes</b>
UC-1	Register Standard User	Click on ‘Sign up’ + ‘First Name’ + ‘Last Name’ + ‘Major’ + ‘School’ + ‘Email’ + ‘Phone Number’ (optional) + ‘Password’ + ‘Confirm Password’ + ‘Agree to terms..’ + ‘Submit’	11
UC-2	Register Premium User	Click on ‘Sign up’ + ‘First Name’ + ‘Last Name’ + ‘Position’ + ‘School/Company’ + ‘Email’ + ‘Phone Number’ (optional) + ‘Password’ + ‘Confirm Password’ + ‘Agree to terms..’ + ‘Submit’	11
UC-3	Account Login	Click on ‘Email/Phone Number’ + ‘Password’ + ‘Captcha’ + ‘Remember Me’ (Optional) + ‘Log in’	5
UC-4	Edit Profile	Click on ‘Account icon’ + ‘Edit’ + (Depend on user’s effort/preference) + ‘Save’	Unclear
UC-5	Post In Forum	Click on ‘Post box’ + ‘Enter the information’ + ‘Add tags’ + ‘Add photos/files’ *Optional) + ‘Post’	5
UC-6	View Forum	Click on ‘Forum’	1
UC-7	Create Discussion Group	Click ‘chat’ + click ‘search person’ + enter to find person + click on correct person + add topic + add message + send	7
UC-8	Join Forum Discussion	Click on ‘Add answer/comment’ + ‘Post’	2
UC-9	Moderate Posts	None	0
UC-10	Upvote/Downvote	Click upvote or downvote	2
UC-11	Post in Opportunity Pages	Click on ‘Post box’ + ‘Enter the information’ + ‘Add tags’ + ‘Add photos/files’ *Optional) + ‘Post’	5
UC-12	View Opportunity Page	Click on ‘Opportunity page’	1
UC-13	Bookmark Posts	Click on ‘Bookmark icon’	1
UC-14	Follow Other Users	Click on ‘Follow’	1
UC-15	Create Private Chat	Click ‘chat’ + click ‘search person’ + enter to find	6

		person + click on correct person + add message + send	
UC-16	Search	Click on search bar + click search	2
UC-17	Filter	Click on 'Filter' + choose available options + select	3
UC-18	Access Email System	Click on email + click on compose + click on receivers + click on topic + click on dialog box + click send	6
UC-19	Access Exclusive Pages	None	0
UC-20	Get Notification	None	0
UC-21	Manage User Account	Login as Admin + Click on manage user + click on a user + click view/add/update/delete + fill in information + submit	6

## 5. Effort Estimation using Use Case Points

User Case Points method provides the ability to estimate the person-hours a software project requires based on its use cases **(UCP)=UUCP \* TCF\*ECF**

### **UUCP:**

Unadjusted Use Case Points (UUCP) measures the complexity of the functional requirements.

### **UAW**

The Unadjusted Actor Weight (UAW), based on the combined complexity of all the actors in the use cases.

Actor Name	Description of relevant characteristics	Complexity	Weight
Student	Student interacts with the system via a graphical user interface	Complex	3
Job seeker	Job seeker interacts with the system via a graphical user interface	Complex	3
Recruiter	Recruiter interacts with the system via a graphical user interface	Complex	3
Admin	Admin interacts with the system via a graphical user interface	Complex	3

Teacher	Teacher interacts with the system via a graphical user interface	Complex	3
Tutor	Tutor interacts with the system via a graphical user interface	Complex	3
Email System	Email system interacts with our system through a network communication protocol.	Average	2
Database	Database interacts with our system through a network communication protocol.	Average	2

$$\text{UAW} = (6 \times \text{complex}) + (2 \times \text{average}) = 6(3) + 2(2) = 18 + 4 = 22$$

### UUCW

The Unadjusted Use Case Weight (UUCW), based on the total number of activities (or steps) contained in all the use case scenarios.

Use Case	Action	Description	Category	Weight
UC-1	Register Standard User	<b>Average user interface.</b> Upon new account registration, user can choose to sign up as standard user. User will be asked to fill in credential informations such as first name, last name, email address, username, and password.	Average	10
UC-2	Register Premium User	<b>Average user interface.</b> Faculties, recruiters, event organizers, employers sign up for a premium account. To do so users are required to provide their first name, last name, email address, phone number, photo of national ID and proof of employment. The registration requires administrator approval.	Average	10
UC-3	Account Login	<b>Simple user interface.</b> Users fill in their username and password to login, and it will direct them to their profile page.	Simple	5
UC-4	Edit Profile	<b>Complex user interface.</b> Users edit their profile information.	Complex	15
UC-5	Post In Forum	<b>Average user interface.</b> Users post any	Average	10

		topic/questions to forum.		
UC-6	View Forum	<b>Average user interface.</b> Users view forum questions, comments, and number of up/downvotes.	Average	10
UC-7	Create Discussion Group	<b>Average user interface.</b> Users create discussion group in order to post information/questions specific to that topic. When creating the group, they have to select its category and make it either public or private. After it's created, it will generate a link that directs you to the discussion page which the admin of the group can share with others. If it's a private discussion, it will require permission from the admin before other user can access it.	Average	10
UC-8	Join Forum Discussion	<b>Average user interface.</b> Users make comments, answer questions, post a question/topic, upvote/downvote other's comments/post.	Average	10
UC-9	Moderate Posts	<b>Complex user interface.</b> Administrator needs to maintain control of the system. Therefore, if any post or comment is reported as inappropriate or irrelevant, administrator has privilege to remove that post and if a user's post/comment is reported more than 3 times, system administrator will have to block that account temporarily to avoid any more suspicious activities.	Complex	15
UC-10	Upvote/Downvote	<b>Simple user interface.</b> User presses the triangle or inverse triangle next to the post or comments in order to upvote/downvote it. System automatically adds points to the user rating if their post/comments get upvoted and subtract the point if otherwise.	Simple	5
UC-11	Post in	<b>Average user interface.</b> Only premium	Average	10

	Opportunity Pages	users can create post in this page. User will be asked to provide a heading, a short description a long description, category tags, and optionally an image describing the opportunity. Then it will wait for the administrator/moderator to approve of the post before it's accessible to the public.		
UC-12	View Opportunity Page	<b>Simple user interface.</b> Opportunity page includes posts about scholarships, university programs, job announcements, internships, competitions, workshops, courses, fairs, and volunteer opportunities. Users can view posts in the page, and they can filter it by tags or category.	Simple	5
UC-13	Bookmark Posts	<b>Simple user interface.</b> Once they see any post they're interested in, users can bookmark it for later view. To access the bookmarked posts, user clicks on the bookmark icon.	Simple	5
UC-14	Follow Other Users	<b>Simple user interface.</b> Users click on subscribe/follow other users to get notification about their posts. And they click on unfollow to stop getting the notification.	Simple	5
UC-15	Create Private Chat	<b>Average user interface.</b> Users reach out and connect to others through chat function, thus it increases the amount of interaction between users. They will first need to go to the other user's profile page and click on the message icon in order to start chat. Alternatively, user can go to their contact book in order to search for the person they want to start chat with.	Average	10
UC-16	Search	<b>Simple user interface.</b> Users search for existing posts, topics, questions through the search bar by inputting keywords.	Simple	5
UC-17	Filter	<b>Simple user interface.</b> Each post, forum question, and topic has tags associated to them to identify the category. Users	Simple	5

		select categories of their interest in order to filter posts.		
UC-18	Access Email System	<b>Average user interface.</b> System enables users to connect with each other through other communication channels such as email as well.	Average	10
UC-19	Access Exclusive Pages	<b>Simple user interface.</b> Only users with predetermined ratings will have access to exclusive pages.	Simple	5
UC-20	Get Notification	<b>Simple user interface.</b> Users get notifications for any account related activities or messages.	Simple	5
UC-21	Manage User Account	<b>Simple user interface.</b> Admins view, add, update, and delete any user account and the user profile information including profile photo, personal information, educational background, work experiences, volunteer experiences.	Simple	5

$$\text{UUCW} = (2*\text{complex}) + (9*\text{average}) + (9*\text{simple}) = (1*15) + (9*10) + (10*5) = 30+90+45=155$$

$$\text{UUCP} = \text{UAW} + \text{UUCW} = 22 + 155 = 177$$

**TCF:**

Technical Complexity Factor (TCF) - Nonfunctional Requirements

Technical Factor	Description	Weight	Perceived Complexity	Calculated Factor (Weight*Perceived Complexity)
T1	Distributed system	2	3	6
T2	Performance objectives	2	3	6
T3	End-user efficiency	1	3	3
T4	Complex internal processing	1	4	4
T5	Reusable code	1	3	3
T6	Easy to install	1	1	1
T7	Easy to use	1	1	1

T8	Portable	2	0	0
T9	Easy to change	1	1	1
T10	Concurrent use	1	4	4
T11	Special security features	1	5	5
T12	Provide direct access for third parties	1	0	0
T13	Special user training	0.5	0	0
Total				34

**TCF=Constant-1 + Constant-2\*Technical Factor Total**

$$\text{TCF} = 0.6 + (0.01 * 34) = 0.94$$

### **ECF**

Environmental Complexity Factor (ECF) measure the experience level of the people on the project and the stability of the project.

<b>Environmental Factor</b>	<b>Description</b>	<b>Weight</b>	<b>Perceived Impact</b>	<b>Calculated Factor (Weight * Perceived Complexity)</b>
E1	Familiarity with development process	1.5	1	1.5
E2	Application problem experience	0.5	2	1
E3	Paradigm experience	1	2	2
E4	Lead analyst capacity	0.5	1	0.5
E5	Motivation	1	4	4
E6	Stable requirement	2	4	8

E7	Part-time staff	-1	0	0
E8	Difficult programming language	-1	3	-3
Environmental Factor Total				14

**ECF=Constant-1 + Constant-2 \* Environmental Factor Total**

$$\text{ECF} = 1.4 + (0.03 \times 14) = 1.82$$

**UCP = UUCP\*TCF\*ECF = 177\*0.94\*1.82= 302.8 use case points**

If we assume we have a productivity factor (PF) of 28:

**Effort estimation (units are person-hours) = 302.8 \*28 = 8478.4 person-hours**

## 6. Domain Analysis

### 6.1. Concept Definition

Concept Name	Type	Concept Definition
Interface Page	D	Displays a page to interact with user, shows user current context and navigation, and render records to user
Login Controller	D	Logs user into the system
Register Controller	D	Register new user account
Authentication Controller	D	Checks if user account exists and authenticate user to the system
Forum Controller	D	Creates forum question record when user post a new question
Opportunity Page Controller	D	Creates an opportunity page post record when user submits an opportunity
User Profile Controller	D	Creates user profile record when user submits profile information
User Controller	D	Creates user account when user signs up
User Model	K	Container for user account information such as username and password
Forum Question Model	K	Container for forum questions

Forum Answer Model	K	Storage for forum answers
User Profile Model	K	Container for user profile information such as education history and employment records
Opportunity Page Post Model	K	Container for opportunity page posts such as post category, post title, post description
Database Connection	D	Sends database query to the database and retrieves the rows
Email Notifier	D	Notifies user of account activities through email

## 6.2 Association Definition

Concept Pair	Association Description	Association Name
Interface Page, Login Controller	Login Controller gets account email and password from user login interface page	Requests data
Login Controller, Authentication Controller	Login Controller checks with Authentication Controller to verify if account exists	Verifies account
Authentication Controller, User Model	Authentication Controller checks in User Model for matching user account	Checks data
User Model, Database Connection	User Model retrieves user record from Database Connection	Requests data
Interface Page, Register Controller	Register Controller gets account information from user registration interface page	Requests data
Register Controller, User Model	Register Controller creates new user record in User Model	Requests data
User Model, Database Connection	User Model saves user record to Database Connection	Requests data
Register Controller, Email Notifier	Register Controller requests email notifier to send	Requests Notify

	confirmation email to user	
Opportunity Page Controller, Opportunity Page Post Model	Opportunity Page Controller requests opportunities from Opportunity Page Post Model	Requests data
Opportunity Page Post Model, Database Connection	Opportunity Page Post Model retrieves post information from Database Connection	Requests data
Opportunity Page Controller, Opportunity Page Comment Model	Opportunity Page Controller requests post comments from Opportunity Page Comment Model	Requests data
Opportunity Page Comment Model, Database Connection	Opportunity Page Comment Model retrieves post comments from Database Connection	Requests data
Opportunity Page Controller, Authentication Controller	Opportunity Page Controller checks user permissions from Authentication Controller if user is eligible to post on the opportunity page	Requests verify
Opportunity Page Controller, Opportunity Page Post Model	Opportunity Page Controller requests list of posts from Opportunity Page Post Model	Requests data
Opportunity Page Post Model, Database Connection	Opportunity Page Post Model saves posts to Database Connection	Saves data
Forum Controller, Forum Question Model	Forum Controller requests questions from Forum Question Model	Request data
Forum Controller, Forum Answer Model	Forum Controller requests answers from Forum Answer Model	Requests data
Forum Question Model, Database Connection	Forum Question Model retrieves questions from Database Connection	Requests data

Forum Answer Model, Database Connection	Forum Question Model retrieves answers from Database Connection	Requests data
Forum Controller, Authentication Controller	Forum Controller checks user permissions from Authentication Controller if user is eligible for posting in forum	Requests verify
User Profile Controller, User Profile Model	User Profile Controller fetches and update user profile to User Profile Model	Requests data

### 6.3. Attribute Definitions

Concept	Attribute	Attribute Definition
Interface Page	Interface page	Displays requested page to user
Register Controller	Data create	Create new account when user register
Authentication Controller	Data verify	Checks to authenticate user
Login Controller		
User Profile Controller	Data create	Get input data to store to the database
User Controller		
Forum Controller		
Opportunity Page Controller		
User Model	Data Storage	Container for data
Forum Question Model		
Forum Answer Model		
User Profile Model		
Opportunity Page Post Model		

Database Connection		
Email Notifier	Notifier	Notify user of account related activities

#### 6.4. Traceability Matrix

		Domain Concepts														
Use Case	Point Weight	Interface Page	Login Controller	Register Controller	Authentication Controller	Forum Controller	Opportunity Page Controller	User Controller	User Profile Controller	User Model	Forum Question Model	Forum Answer Model	User Profile Model	Opportunity Page Post Model	Database Connection	Email Notifier
UC-1	59	X		X										X	X	
UC-2	59	X		X			X		X					X	X	
UC-3	52	X	X		X			X	X					X		
UC-4	41	X						X	X			X		X		
UC-5	54	X				X					X	X			X	
UC-6	55	X				X					X	X			X	
UC-7	49	X				X					X	X			X	
UC-8	45	X			X	X					X	X			X	
UC-9	45	X									X	X			X	
UC-10	47	X				X					X	X			X	
UC-11	52	X			X		X							X	X	
UC-12	53	X					X							X	X	
UC-13	44	X					X	X	X					X	X	
UC-14	42	X						X		X			X		X	
UC-15	42	X													X	
UC-16	42	X													X	
UC-17	42	X													X	

UC-18	39	X												X	X
UC-19	41	X												X	
UC-20	33	X												X	X

The traceability matrix above is based upon the traceability matrix of use cases that were derived from the software requirement. We have User Controller that controls user account, Login Controller that manages user login, Register Controller that controls new user account registration, Authentication Controller that verifies user account authentication, Forum Controller that controls forum activities, and Opportunity Page Controller that controls posts in opportunity pages. Each of the Model stores specific data in the system. The system also consists of Email Notifier that sends notifications related to user account activities or new user account confirmation.

## 7. Interaction Diagram

Interaction diagram shows the responsibility of system objects and how they interact with each other. Below are the interaction diagrams of the 6 use cases specified in the fully dressed description in report 1.

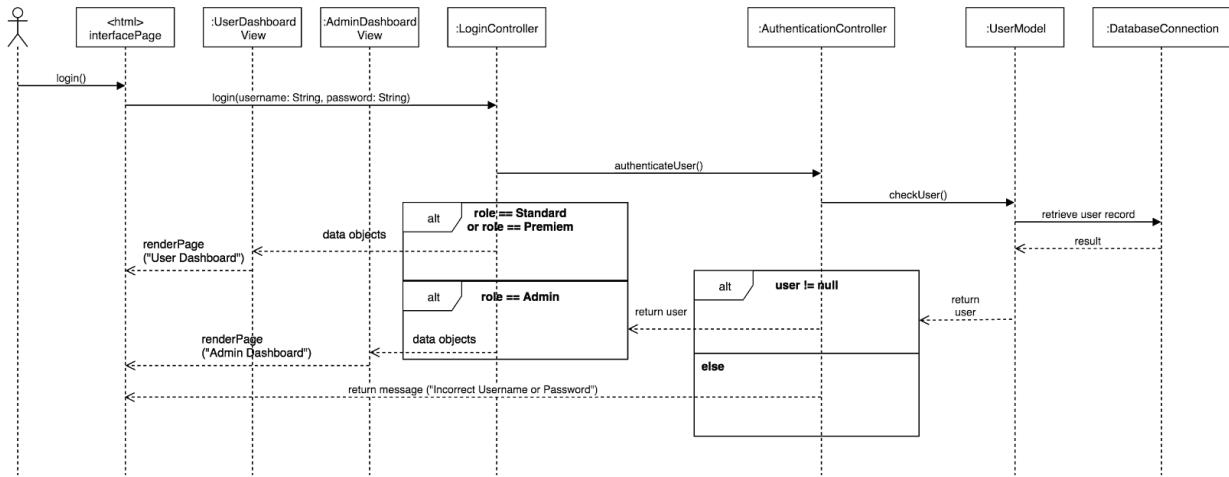
### **Design Architecture**

It's worth noting that nEdCom application is built in laravel, a PHP framework that uses MVC (Model, View, Controller) architecture. View layer is responsible for user interface pages and displaying object data to the user in an appropriate format. Controller takes requests from the user, performs computations and logic, and returns objects to View. Model retrieves and send data to the database as needed by the Controller.

### **Design Principle**

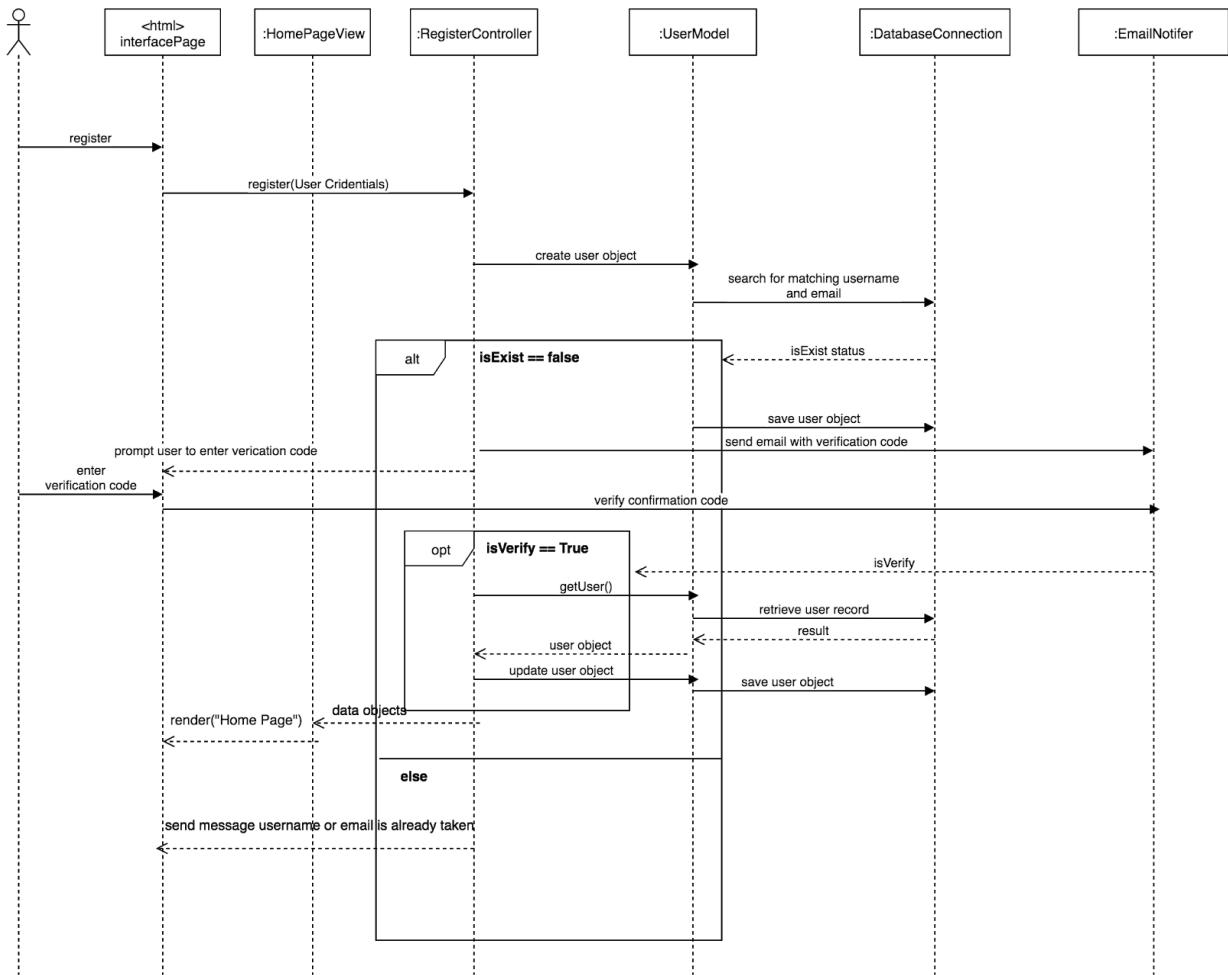
- Expert Doer: View layers such as UserDashboardView, AdminDashboardView, OpportunityPageView, ForumPageView focus rendering html pages. Authentication controller focuses on authenticating the user. Model layers such as UserModel, OpportunityPagePostModel, ForumQuestionModel, ForumAnswerModel focuses on retrieving records from database.
- High Cohesion: Each object is assigned specific computation responsibility. Example: LoginController checks role in user object and call the corresponding view layer file accordingly to render the html page.
- Low Coupling: Each object does not take on many communication responsibilities. For example, Authentication controller only communicates with LoginController and UserModel. UserModel only communicates with database and controller.

## Login



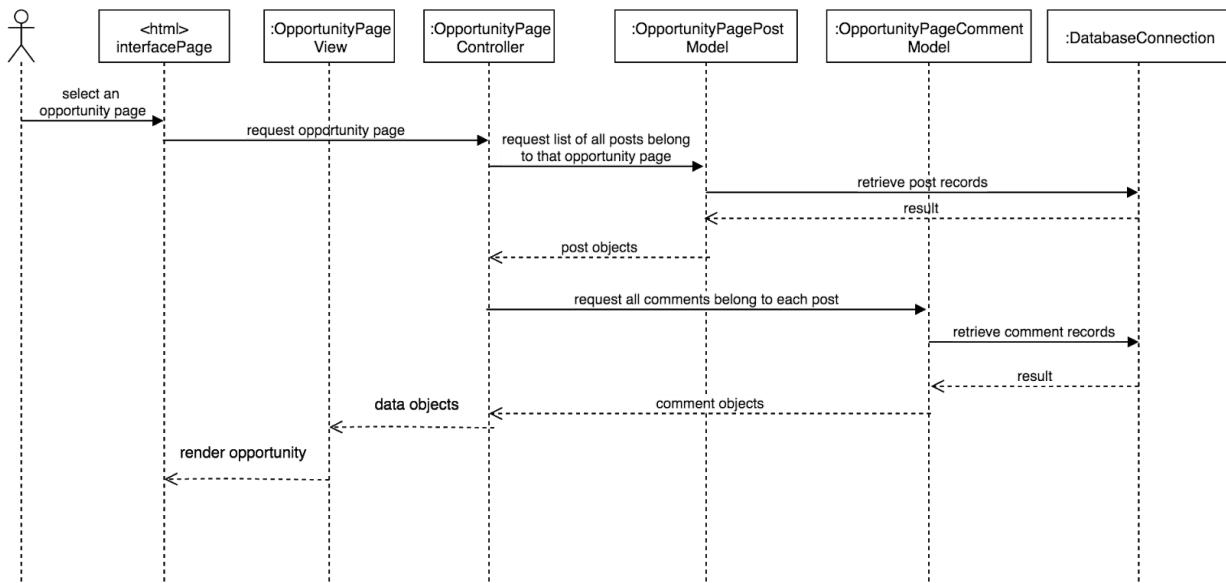
- The login function with username and password as input parameters in loginController is called when the user inputs the credentials and clicks on login.
- The loginController then calls the authenticateUser() function in AuthenticationController to authenticate the user.
- The AuthenticationController then calls the UserModel to find user with the given credential. The UserModel makes call to database to retrieve user record that matches the credential.
- The user object is returned to the AuthenticationController.
  - If the user object is not null, that means the user exists in our database. Therefore, the user object is returned to the LoginController.
    - Once the LoginController receives the user object, it checks for user role.
    - If user role is ‘standard’ or ‘premium’, it makes call to UserDashboardView to render the User Dashboard html page for user to see
    - If user role is ‘admin’, it makes call to AdminDashboardView to render the Admin Dashboard html page for user to see.
  - However, if the user object is null, that means the user does not exist in our database. The message “Incorrect Username or Password” is shown to the user.

## Registration



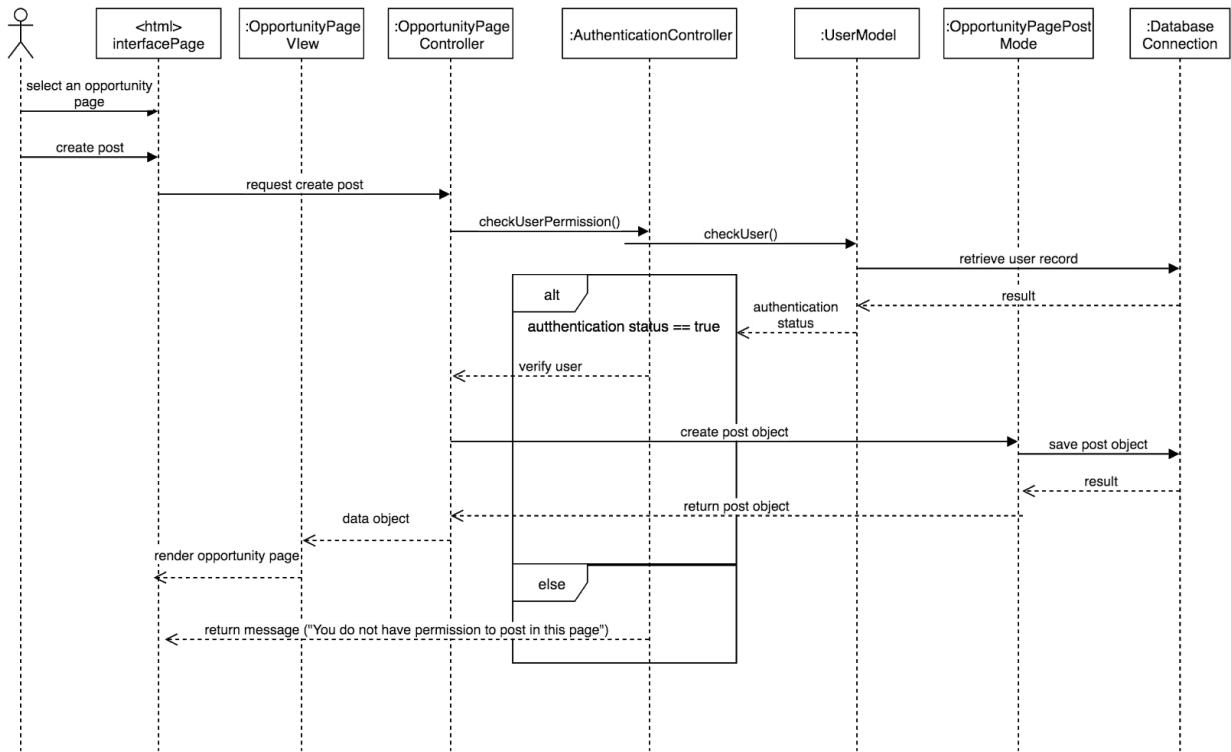
- RegisterController creates user object with the input information and pass it to UserModel. The user model then search for matching username and email to see if they are already taken.
  - If username and email are not taken. The UserModel saves the user object in the database. The RegisterController then requests the EmailNotifier to send verification code via email to the user. After the user submit the correct verification code, their verification status in the database will be updated.
  - If username or email are already taken, the message “Username or email are already taken” is shown to the user.

## View Opportunity Page



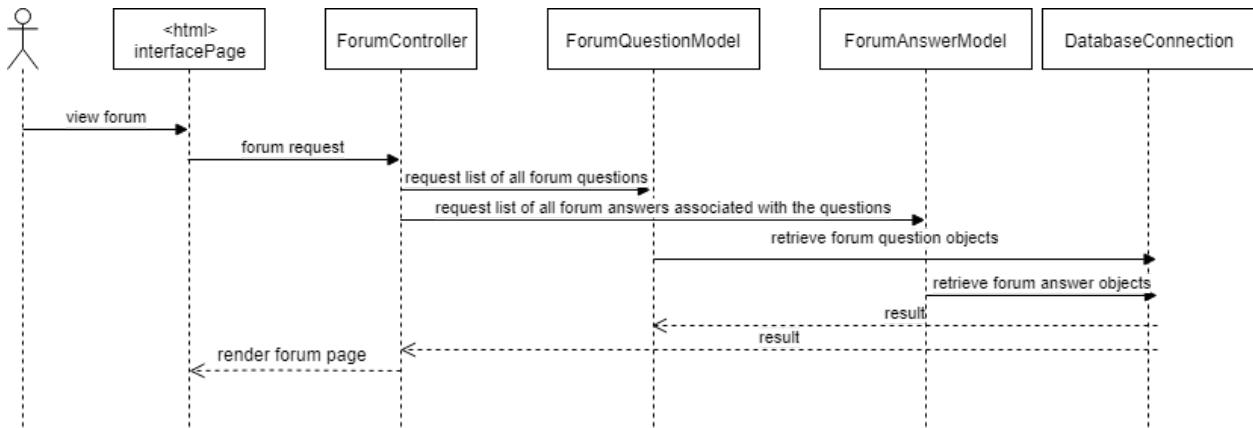
- Once the user selects on a specific opportunity page (competitions, scholarships, internships, career, or volunteering opportunities), the OpportunityPageController make calls to OpportunityPagePostModel to get all the posts belong to that opportunity page. OpportunityPageController also makes call to OpportunityPageCommentModel to get all comments belong to the each of the post. The two models retrieve the requested data from the database and return them to OpportunityPageController. The controller then returns the data to the OpportunityPageView to render the html page for user to see.

## Post in Opportunity Page



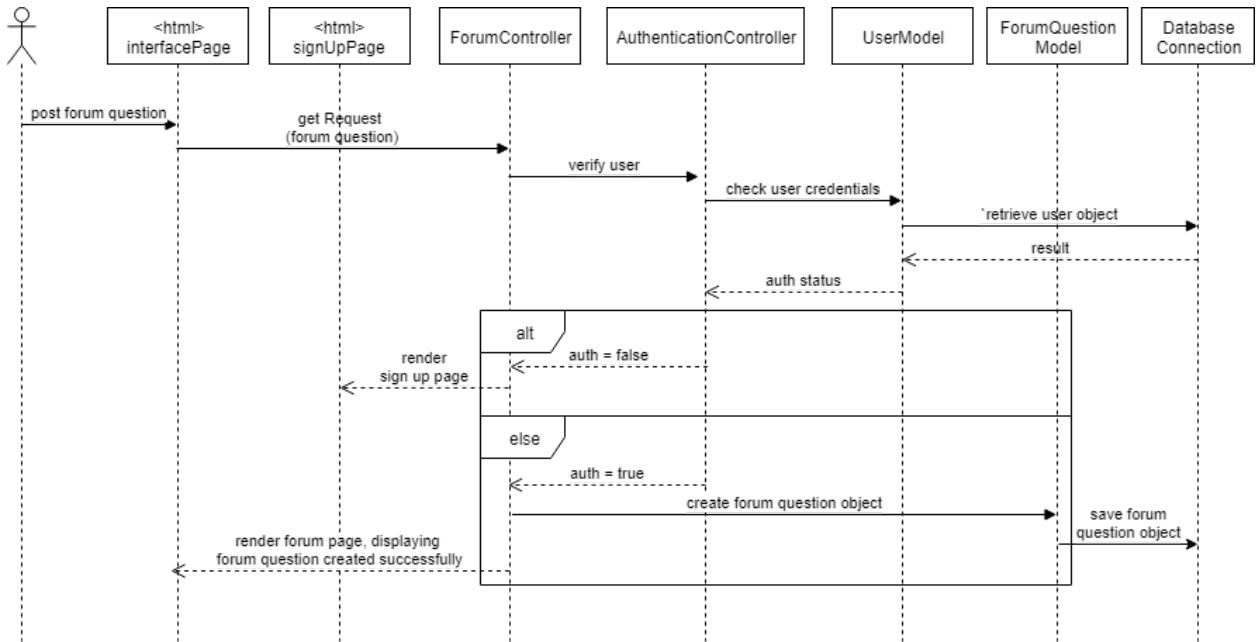
- Once the user selects a specific opportunity page (competitions, scholarships, internships, career, or volunteering opportunities) and select create post, the OpportunityPageController will make calls to the AuthenticationController to check whether the user has permission to post or not.
  - If the user has permission to post, the OpportunityPage creates the post object and let the OpportunityPagePostModel save the record in the database. After that, the OpportunityPageController returns the successfully created post to the OpportunityPageView to render the html page for user to see.
  - If the user does not have permission to post, the message “You do not have permission to post in this page” is shown to the user.

## View Forum

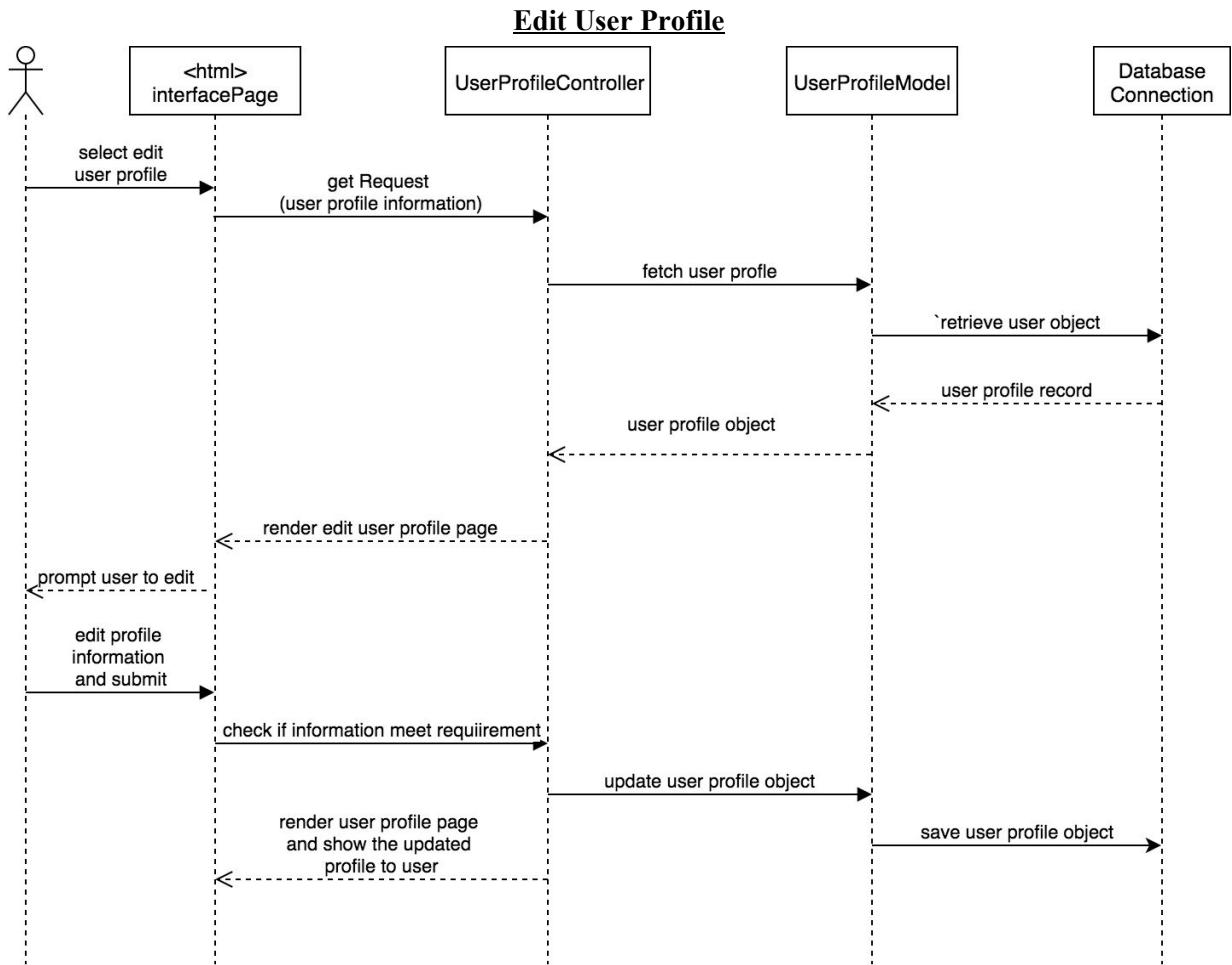


- Once the user opens the forum page, the `ForumController` make query call to the `ForumQuestionModel` to request list of all forum questions and the `ForumAnswerModel` to request list of all forum answers associated with each forum questions. Then both the models retrieve the data from the database. The database return results as queried, and the `ForumController` take the data objects to display to the forum page.

## Post in forum

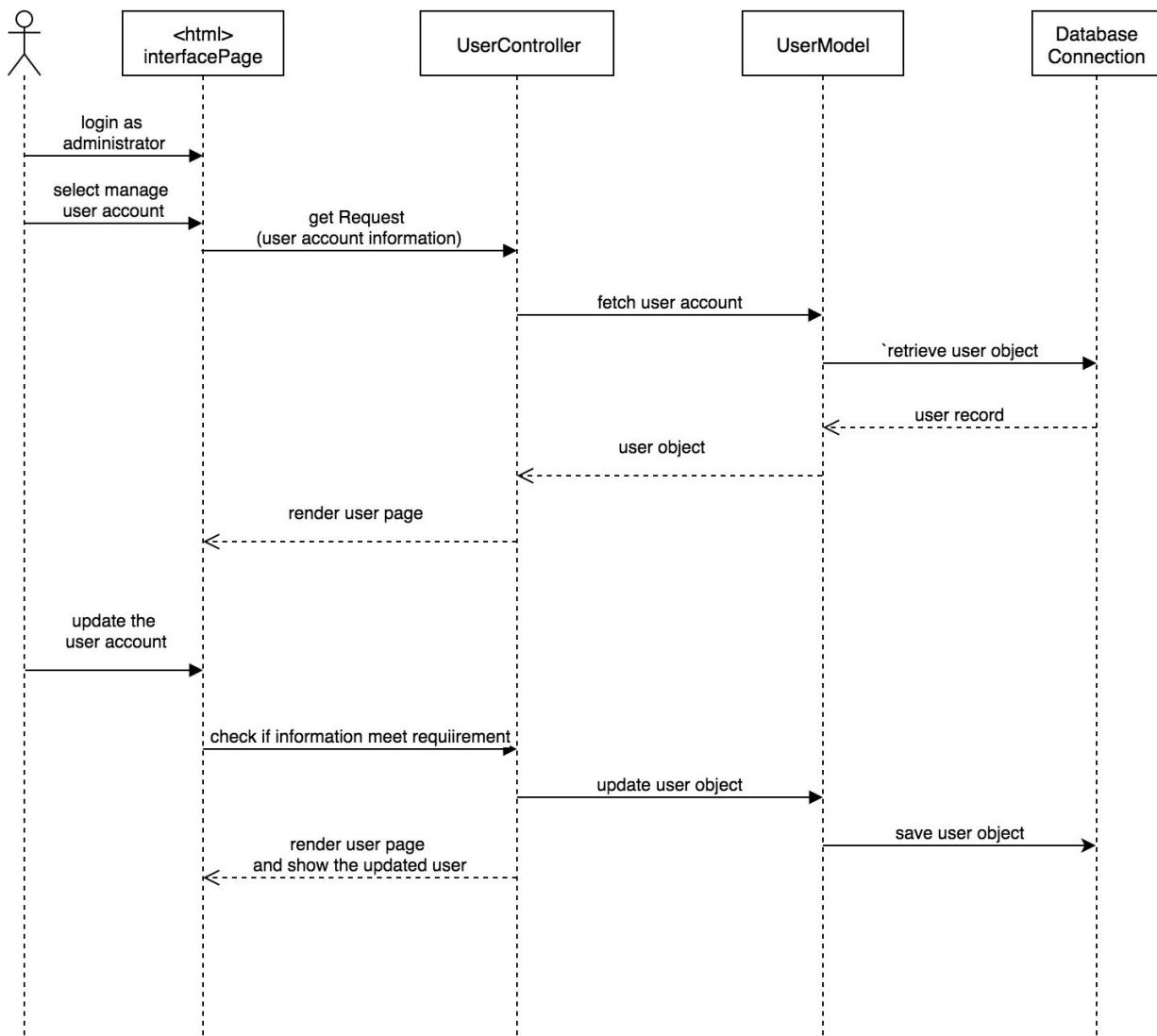


- When user submits a forum question, the ForumController obtains the input data from the interface page, then checks with the AuthenticationController to verify the user. The AuthenticationController asks the UserModel if user already has an account. The UserModel take the request and retrieve user object from the database. Then the database return the result as queried. The UserModel take the result and return the user's auth status to the AuthenticationController.
  - If auth is false, user doesn't have an account yet. The ForumController will render the signup page for user to register for an account.
  - If auth is true, user already has an account and is eligible to post forum question. ForumController takes the input data and request ForumQuestionModel to create new forum question object which is saved to the database. Then ForumController renders the forum page so that user can see the forum question they have posted.



- Once the user has logged into his/her own account and navigate to user profile, the user information will be displayed along with “edit” button. The user can click on edit and the “edit user profile” page will be displayed along with the user profile information in editable input boxes. The user can edit his/her own profile information and then submit the changes. The system will verify if the submitted information meet the requirement. If the information meets the requirement, the user profile object will be updated and saved. Then the user profile page with the updated profile information will be rendered and displayed to user.

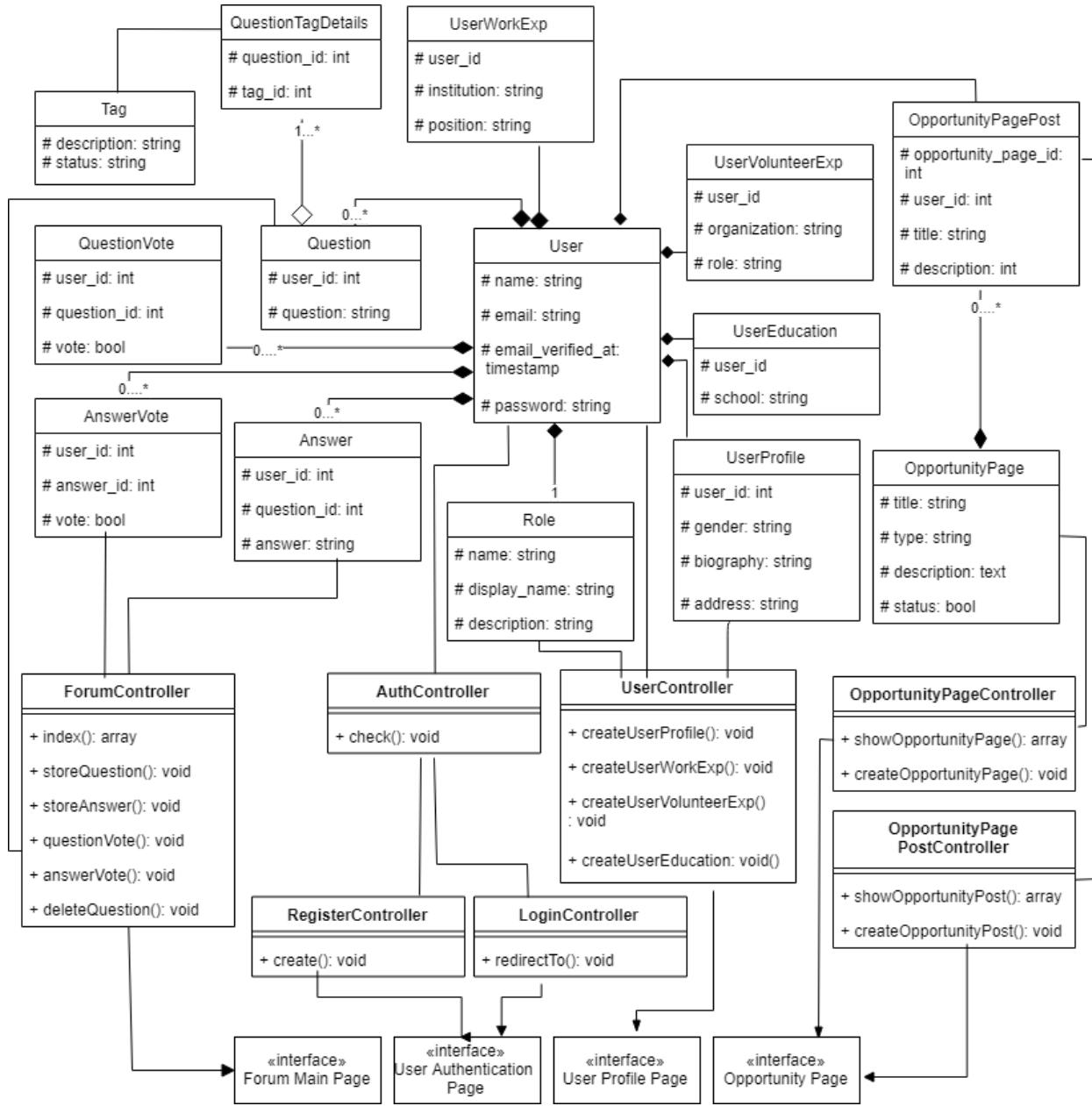
## Manage User Account (Update User Account)



- The user first need to login as administrator with administrator account.
- Once the user is logged in as administor, the user can navigate to “Manage User Account” page. From there, a list of all users who are registered on the platform are displayed on the page. The admin can then select a specific user from the page and choose to edit the user account. The UserController will verify if the request to update the user account contains valid information. If the information is valid, the user object is updated and saved. The newly updated user’s information is displayed to the user.

## 8. Class Diagrams And Interface Specification

### 8.1. Class Diagram



## 8.2. Data Types And Operation Signatures

**User:** a class that contains user account information

- Attributes:  
# string name  
# string email  
# string password

**Role:** a class that contains details associated with user roles.

- Attributes  
# string name  
# string display\_name  
# string description

**UserProfile:** a class that keeps details of user profile information that displays in user profile page

- Attributes  
# int user\_id  
# string gender  
# string address  
# string biography

**Question:** a class that keeps details of forum questions

- Attributes  
# int user\_id: id associated with user who post the question; related to user.user\_id  
# string question

**Tag:** a class that contains all forum tags

- Attributes  
# string description  
# string status: tag status, active/inactive

**QuestionTagDetails:** a class that matches forum question and tags

- Attributes  
# int question\_id: id associated with forum question; related to question.question\_id  
# int tag\_id: id of tag; related to tag.tag\_id

**Answer:** a class that keeps details of forum answers

- Attributes  
# int user\_id: id of user who posted the answer; related to user.user\_id  
# int question\_id: id of question to which the answer belongs; related to question.question\_id  
# string answer

**QuestionVote:** a class that keeps upvote/downvote details for forum questions

- Attributes  
# int question\_id: id of the question that gets voted  
# int user\_id: id of user who clicks upvote/downvote on the question

```
# bool vote: vote status; 1 if upvote; 0 if downvote
```

**AnswerVote:** a class that contains upvote/downvote details for forum answers

- Attributes

```
# int answer_id: id of the answer that gets voted
```

```
# int user_id: id of user who clicks upvote/downvote on the answer
```

```
# bool vote: vote status; 1 if upvote; 0 if downvote
```

**OpportunityPage:** a class that contains information about opportunity page

- Attributes

```
# string title
```

```
# string type
```

```
# string description
```

```
# string status
```

**OpportunityPageController:** a class that handles opportunity page operations

- Operations:

```
+ array showOpportunityPage(): return list of all opportunity pages to the interface page
```

```
+ void createOpportunityPage(): create a new opportunity page and inserts to the database
```

**ForumController:** a class that handles forum operations

- Operations:

```
+ array index(): function that return list of all forum questions to display in forum page
```

```
+ void storeQuestion(): function to store question details
```

```
+ void storeAnswer(): function to store answer details
```

```
+ void questionVote(question_id: int): function to store user votes associated with the forum question with the given id
```

```
+ void answerVote(answer_id: int): function to store user votes associated with the forum answer with the given id
```

```
+ void deleteQuestion(question_id: int): function to delete forum question associated with the given id
```

**UserController:** a class that handles user profile operations

- Operations

```
+ void createUserProfile(): function to create user profile information
```

```
+ void createUserWorkExp(): function to create user work experience information
```

```
+ void createUserVolunteerExp(): function to create user volunteer experience information
```

```
+ void createUserEducation(): function to create user education information
```

**RegisterController:** a class that handles user account registration

- Operations

```
+ void create(): function to register new user account
```

### 8.3. Traceability Matrix

	Classes												
Domain Concepts	User	Role	UserProfile	Question	Answer	QuestionVote	AnswerVote	Tag	OpportunityPage	OpportunityPageController	ForumController	UserController	RegisterController
Database Connection	X	X	X	X	X	X	X	X	X				
User Account Storage	X												
User Profile Storage		X	X										
Forum Question Storage				X									
Forum Answer Storage					X								
Point System						X	X						
Forum Tag Storage								X					
Opportunity Page Storage									X				
Opportunity Page creator										X			
Forum Question Creator											X		
User Profile Creator												X	
User Account													X

Creator												
---------	--	--	--	--	--	--	--	--	--	--	--	--

The diagram traces how the classes evolve from the domain concept model. Some domain concepts that have low priority weight in requirement have not yet been implemented as a class and to be deferred for future implementation.

Model classes made to be correspond with the concept model includes User, Role, UserProfile, Question, Answer, and the OpportunityPage. For the point system, it requires classes to be split into QuestionVote and AnswerVote to keep track of votes (points) user obtains from both their forum questions and answers.

#### 8.4. Object Constraint Language (OCL)

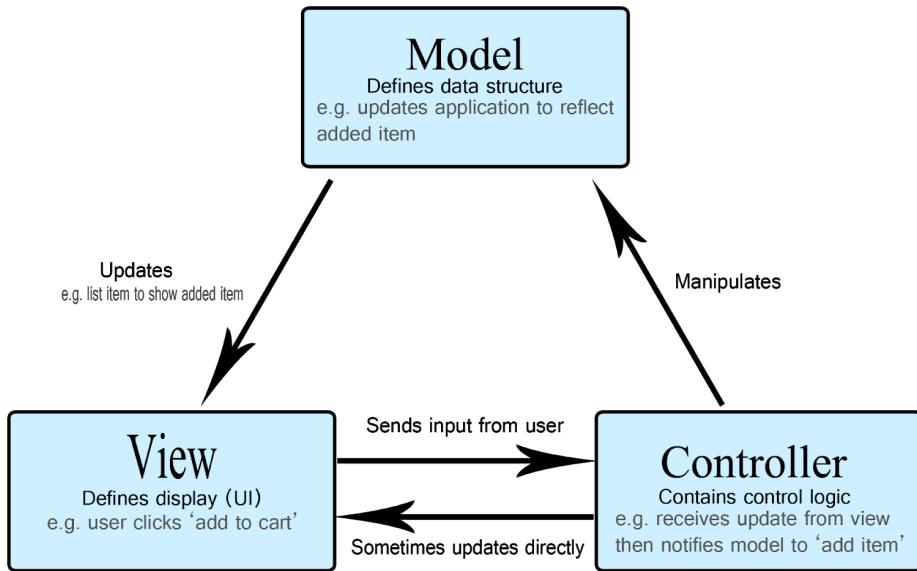
Class	Invariant	Pre-Condition	Post-Condition
User	Database is always available for access	Contains user account information	Display user information
Role	Database is always available for access	Contains details associated with user roles	Get user role and display on user profile page
UserProfile	Database is always available for access	Keep details of the user profile information	Display information in user profile page
Question	Database is always available for access	Keeps details of forum questions	Display questions
Answer	Database is always available for access	Keeps details of forum answers	Display answers
QuestionVote	Database is always available for access	Keeps details of upvote/downvote details for forum questions	Show upvote/downvote details
AnswerVote	Database is always available for access	Contains upvote/downvote details for forum answer	Show upvote/downvote details

Tag	Database is always available for access	Contain forum tag	Display forum questions according to tag
OpportunityPage	Database is always available for access	Contain information about opportunity page	Display opportunity information
OpportunityPageController	Database is always available for access	Handle opportunity page operations	Get and send information to display on opportunity page
ForumController	Database is always available for access	Handle forum operations	Get and send forum data to display on the forum page
UserController	Database is always available for access	Handle user profile operations	Get and send user data
RegisterController	Database is always available for access	Handle user account registration	Get input from registration page and store in database

## 9. System Architecture And System Design

### 9.1. Architectural Styles

nEdComm web application is implemented using Model-View-Controller (MVC) architectural pattern. As the name itself suggests, the MVC architectural pattern separates the application logic into three layers such as Model layer, View layer, and Controller layer.



Source: [developer.mozilla.org](https://developer.mozilla.org)

The View layer defines the presentation of data. It is responsible for rendering user interface pages and displaying data objects to the users in an appropriate format. The Controller defines logics that update the model or view in response to user interaction. It takes requests from the user, performs computations and logic, and returns data objects to View layer. The Model layer defines the data structure of the application. It retrieves data from database and persist data to the database. It transfers data to and from controller as object instances.

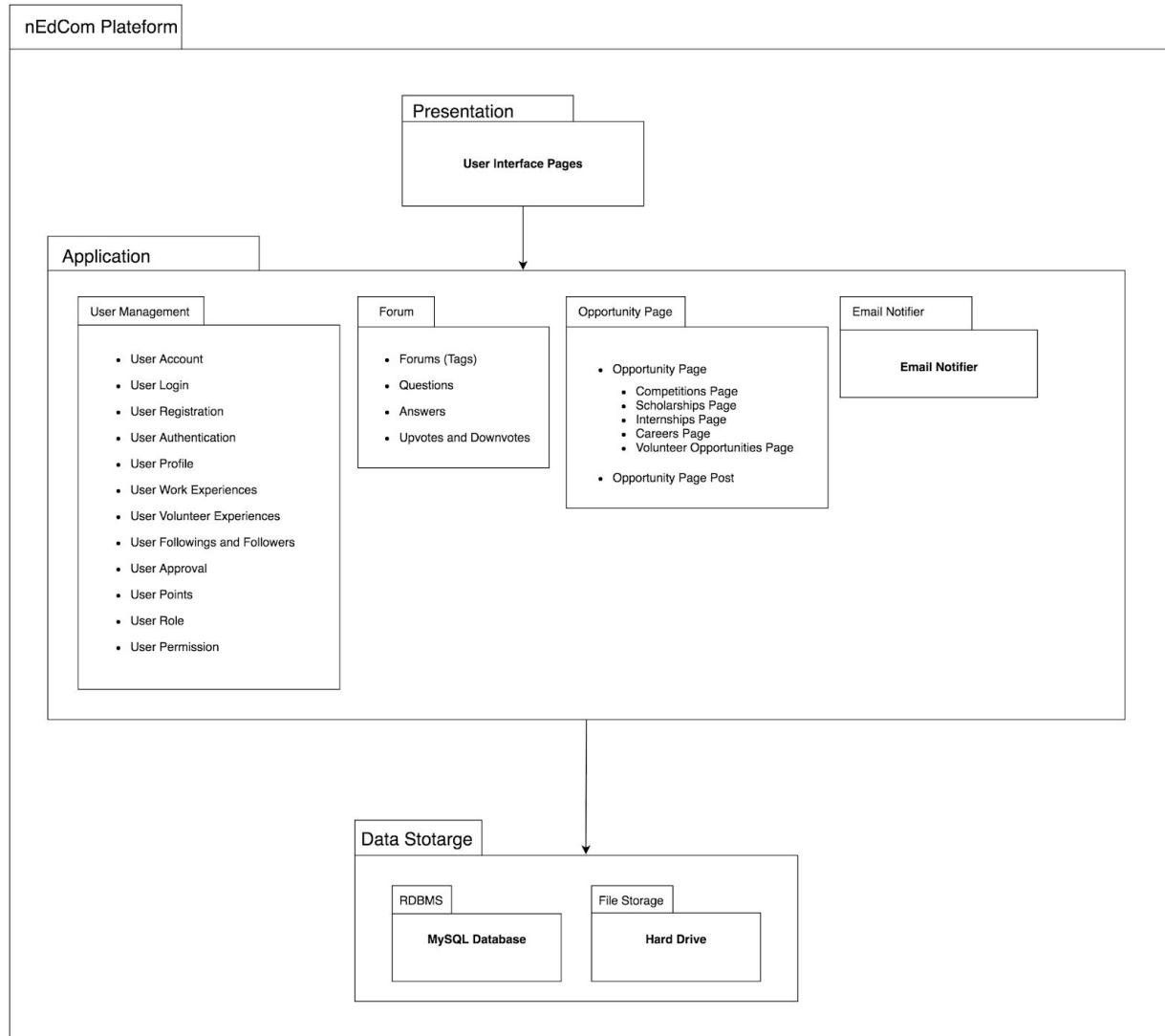
In addition to the Model, View, Controller layers, we also have additional components such as Router, Database Migration, Database Seeder, and Database Factories in building this applications. Router is used for managing all the URLs (Universal Resource Locator) in our application. Database Migration is used for defining database schema and creating all the tables in our database. Database Seeder is used for add initial set of data or testing data into our database. In our case, we use Database Seeder to insert user roles such as 'standard', 'premium', and 'admin' to our database so we can use one of the role when creating new user. Database Factory is used for generating large records of dummy data into our database.

This architectural pattern enable our code base to be modular and reusable. Moreover, it promotes good design principles such as Expert Doer (each layer is expert in their responsibilities), High Cohesion (each controller is assigned specific computation responsibility), Low Coupling (each component does not take on many communication responsibilities). Additionally, this architectural pattern enables software engineer to efficiently split roles for developer in the project and enable them to work simultaneously. Backend developers can work on database schema in Model layer and programming the application and business logic in the controller layer. At the same time, frontend developers can work on designing and programming the user interface in the view layer.

## 9.2. Identifying Subsystems

Our system contains the following subsystems:

- User Management is used for managing user account, user profile, user activities on the platform.
- Forum is used for managing all the forums page (tags), all the posts/questions by users in each of the forums as well as all the answers/comments to posts/questions. It also manages all the upvotes and downvotes made by users to posts/questions and answers/comments.
- Opportunity Page is used for managing all type of opportunity pages such as Competitions Page, Scholarships Page, Internships Page, Careers Page , Volunteer Opportunities Page as well as all the posts in each of those opportunity page.
- Email Notifier is used for sending verification code to users via email,



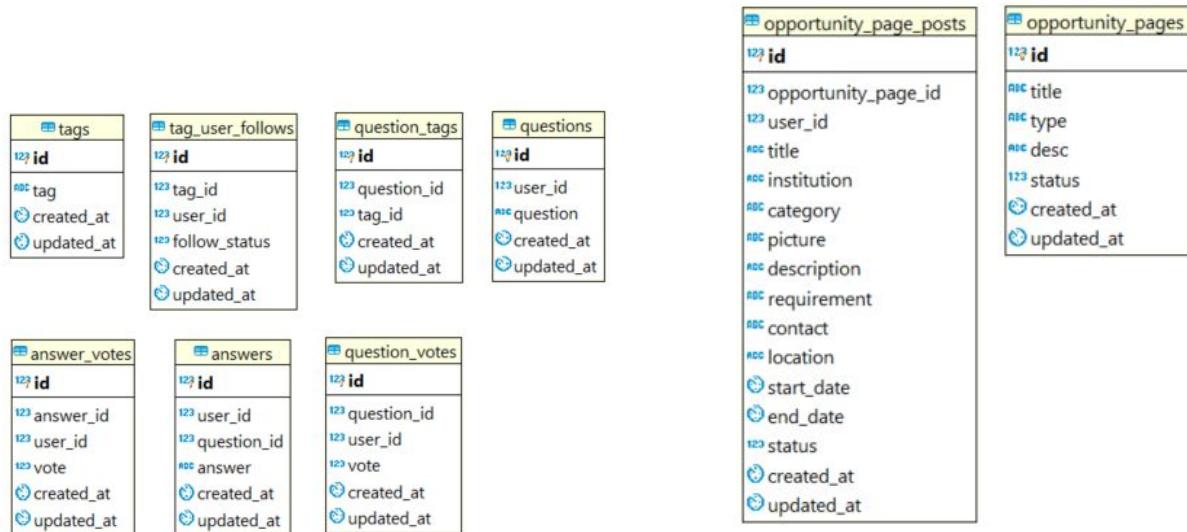
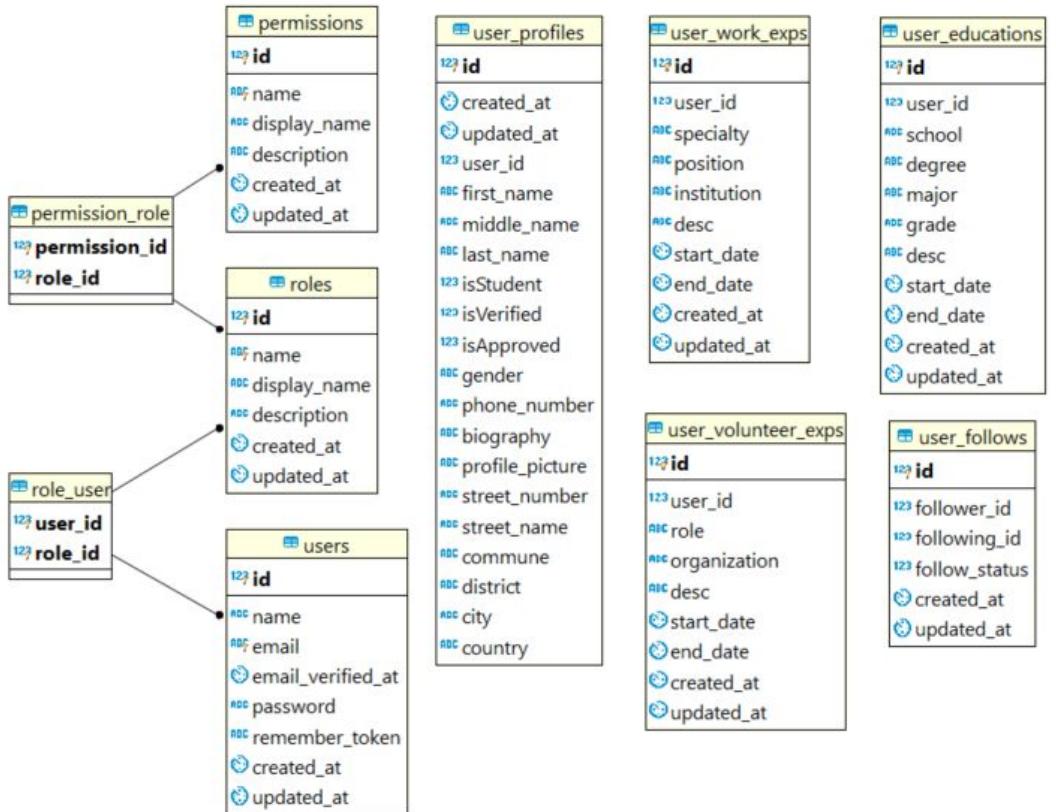
### 9.3. Mapping Subsystems to Hardware

All subsystems except Email Notifier will be running on the Apache Web Server that runs on a Linux Server. For Email Notifier subsystem, it will be implemented through a third party email service API (Application Programming Interface) such as [mailgun](#) that will be run on a server separate from our main server.

### 9.4. Persistent Data Storage

In term of storage, all textual data are stored in MySQL relational database which runs on the same Linux server as the application. All the photos such as JPEG, PNG, GIF and other multimedia formats are stored as files on the same Linux Server as the main application.

## Database Schema



### 9.5. Network Protocol

Our Application uses HTTP (Hypertext Transfer Protocol) to enable communication between clients and server. This network will allow request and response to be send back and forth between client

(user internet browsers such as Chrome, Firefox, Edge, Safari,..etc running on devices like Smartphone, Computer, Tablets...etc) and server (nEdCom laravel web application). HTTP methods such as GET, POST, PUT, DELETE are used.

#### 9.6. Global Control Flow

- a. Execution Orderliness: The system is event driven because it performs certain actions in response to the user interaction or input.
- b. Time Dependency: The system is designed to be real time. When users input information into our platform (for example, posting a question in a forum), those information are recorded into the database immediately. However, user needs to refresh their browser in order for the system to retrieve the latest information and display to them.
- c. Concurrency: Multiple threading will be implemented in future version of the platform especially when we scale the platform to support large number of users accessing the platform simultaneously.

#### 9.7. Hardware Requirements

- a. For client side, End Users need to have electronic devices capable of running modern internet browsers in order to use nEdComm platform.
- b. For server side, nEdComm will be running on a Linux server
  - i. At testing stage, the server needs to have at least
    - 1GB of Memory
    - 20GB of Storage Space
    - High speed Internet connection with large bandwidth
    - 2 Core CPU
  - ii. At production stage, the server needs to have at least
    - 8GB of Memory
    - 100GB of Storage Space
    - 4 Core CPU
    - High speed Internet connection with large bandwidth

## 10. Algorithm And Data Structures

### 10.1. Algorithms

The system implements the following algorithms:

- User Points: Point is a critical part in our platform for determining the credibility of each user. We want to encourage users to actively engage in our platform so system award 1 point to the user when the user's post/comment get upvoted by other users. At the same time, we do realize that sometime user make mistake, so we don't want to penalize too much point when the user's post/comment get downvoted. Thus, we only deduct 1 multiple by weight (in this case 0.1) point when that happens. The algorithm works the following way:

```
if user_post get upvoted
    add 1 point to the user
if user_post get downvoted
    minus (1 x 0.1) point from the user
```

The algorithm is the same for user\_comment.

For simplicity, even though points are stored with all digits in our database, when displaying the point to user, we show them the number that is rounded to the nearest integer, eg. 1, 2, 3, 4.

- Permission to access each page based on User Points
  - To provide incentive for users to acquire more points on our platform, some exclusive opportunity pages and forum pages also have minimum point for entry. Therefore, before allowing user to view each of those pages, we check the following condition first:

```
if user_point >= page_minimun_entry_point
    grant access to user
else
    deny access
```
- Encryption: Our system use Bcrypt hashing algorithm (provided by Laravel) to hash users' password before recording them into the database.
- Serialization: We also serialization algorithm such as toArray and toJSON (built-in with Laravel) to convert Model objects to array or json format respectively.
- Sort: For sorting records, the system uses mysql ORDER BY clause to sort the records ascending or descending.  
Ex: SELECT FNAME, LNAME FROM USER ORDER BY LNAME DESC;
- Search: For searching, the system mysql WHERE clause to select records that match specific criteria.

EX: SELECT FNAME, LNAME FROM USER WHERE FNAME like ‘%am%’ AND LNAME = ‘Cruise’;

## 10.2. Data Structures

We use data structures such Arrays, Object, Stack, Set, Map, all of which are provided natively by PHP and Laravel. On the database side, we store data in a form of rows and columns inside tables of mysql relational database with primary keys and/or foreign keys in each of the table.

# 11. User Interface Design And Implementation

## 11.1 User Interface:

### 11.1.a. Homepage

Homepage interface for login and non-login users is the same. The difference is only when users want to interact with posts; those includes comment, vote, report, post question or other opportunities. In order to get access to those functions, non-login users will be asked to login (if already have an account) or sign up (if no account).

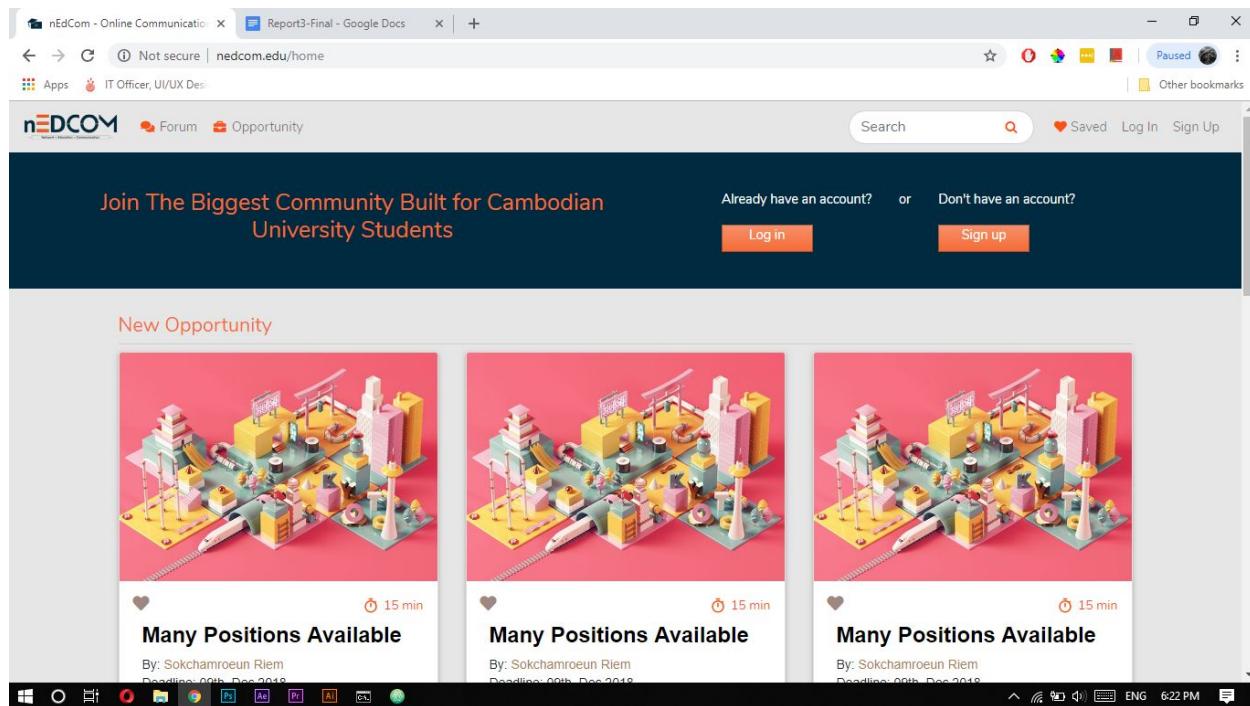


Figure 11.1.a(1): Homepage

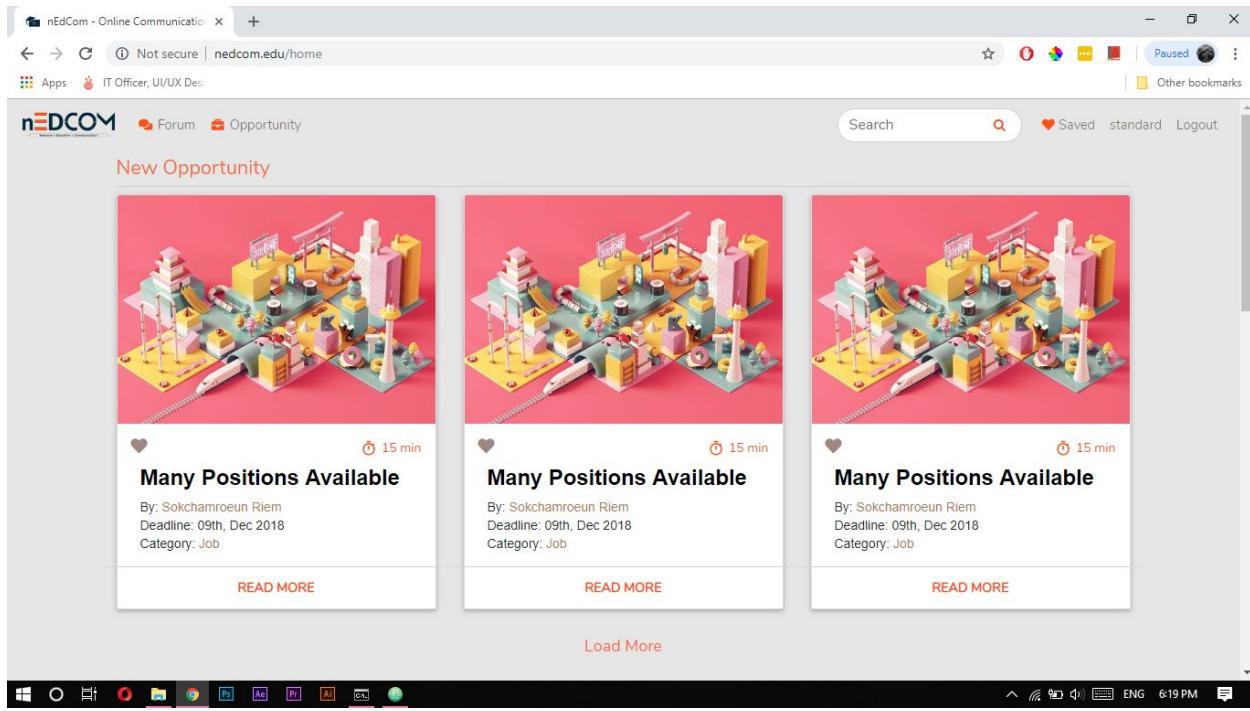


Figure 11.1.a(2): Homepage

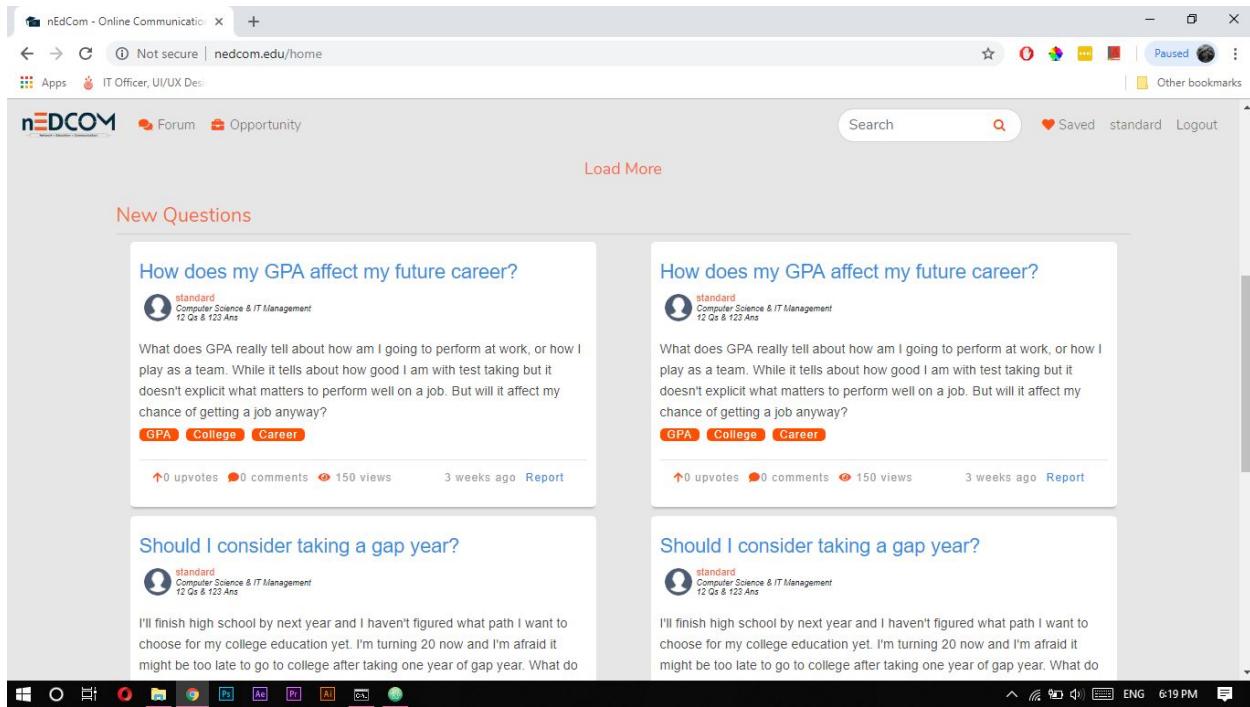


Figure 11.1.a(3): Homepage

### 11.1.b. Signup Pages

- Standard User

The screenshot shows the 'Apply as a Student' registration form. It includes fields for First Name, Last Name, Email, Phone (optional), Major (dropdown with 'Please select your major' and 'Add Now!' link), University (dropdown with 'Please select your university' and 'Add Now!' link), Password, Confirm Password, and gender selection (Male/Female). A checkbox for accepting terms and conditions is present. Navigation links for 'Student' and 'Professor/Hire' are at the top right.

Figure 11.1.b. Sign up page for standard users

#### - Premium User

The screenshot shows the 'Apply as a professor/recruiter' registration form. It includes fields for First Name, Last Name, Email, Phone, Position (dropdown with 'Please select your position' and 'Add Now!' link), Specialty (dropdown with 'Please select your specialty' and 'Add Now!' link), Workplace (dropdown with 'Please select your workplace' and 'Add Now!' link), Password, Confirm Password, and a file upload field for confirming identity ('Choose File'). A note states 'Your privacy is save with us.' A checkbox for accepting terms and conditions is present. Navigation links for 'Student' and 'Professor/Hire' are at the top right.

Figure 11.1.b(2): Sign up page for standard premium users

#### 11.1.c. Login Page

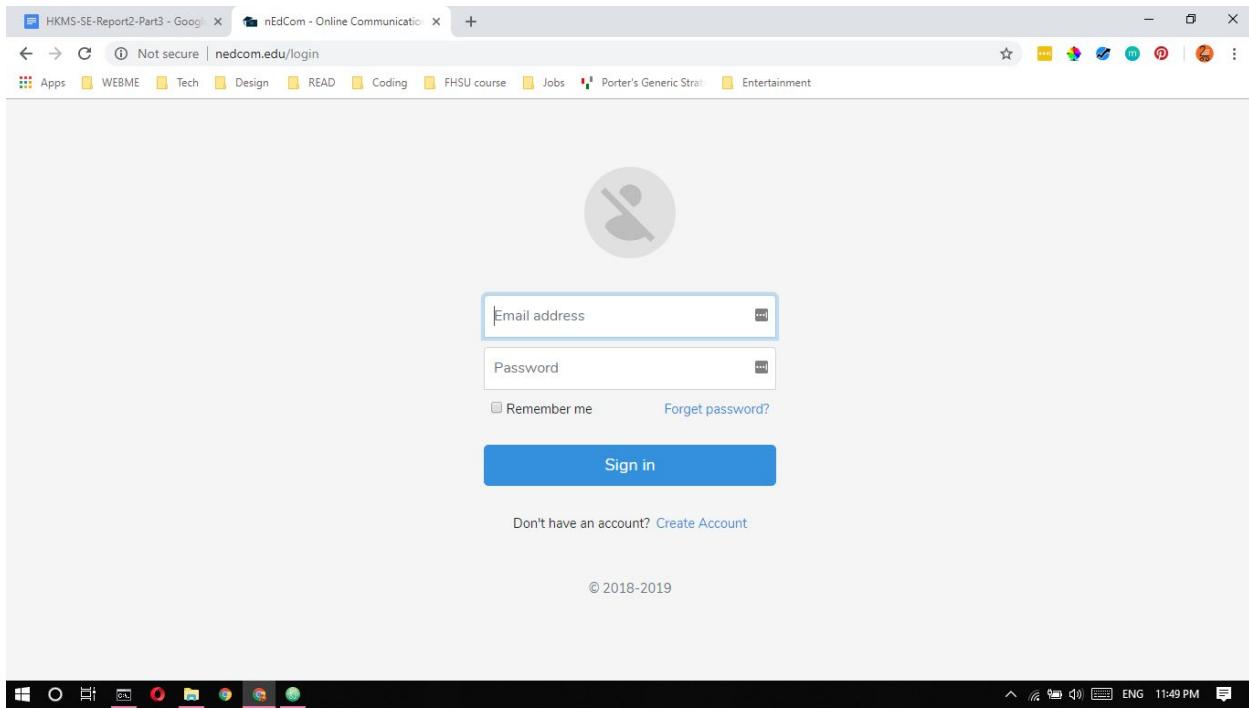


Figure 11.1.c: Login page for all users

#### 11.1.d. Forum homepage

Figure 11.1.d: Forum homepage

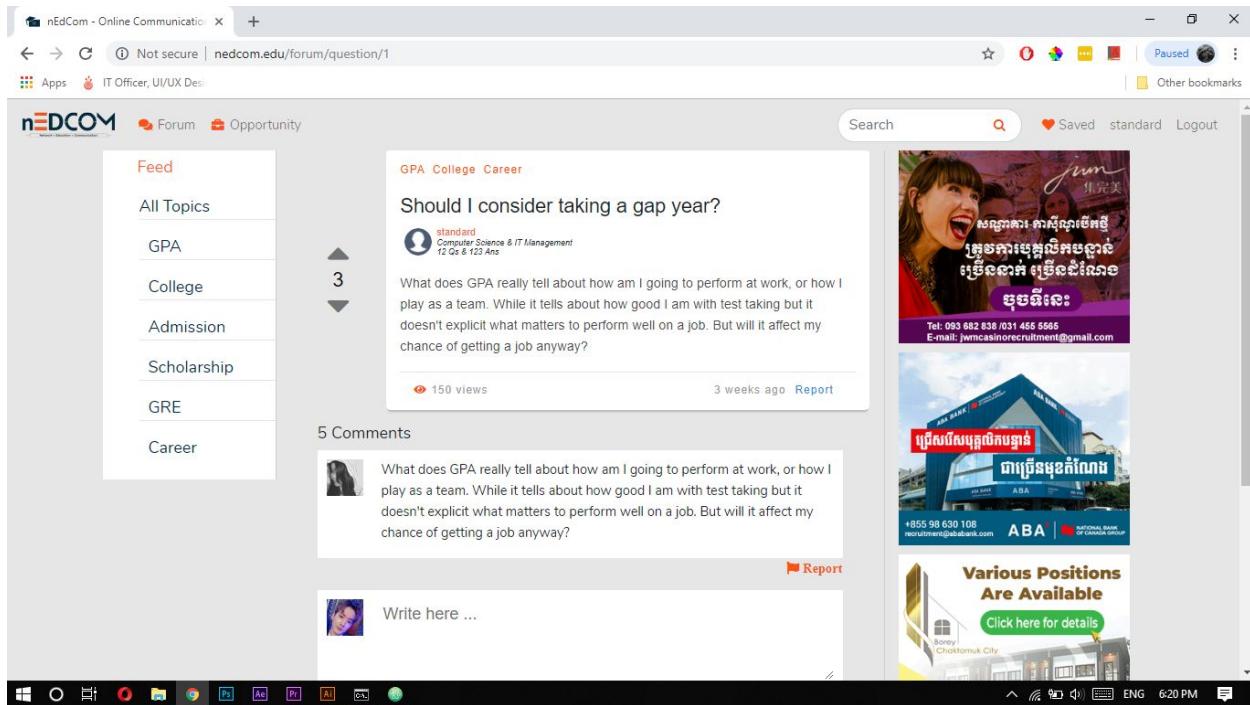


Figure 11.1.d: Forum Detail

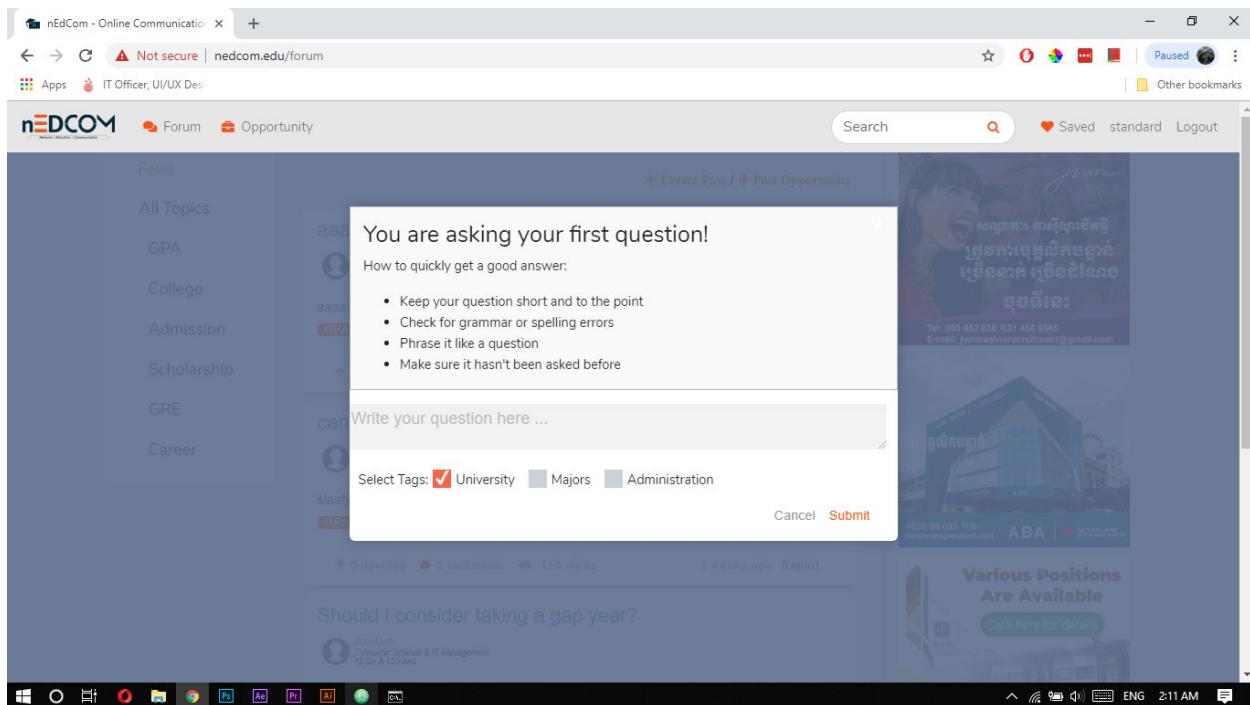


Figure 11.1.d: Forum Create

## 11.1.e. Opportunity homepage

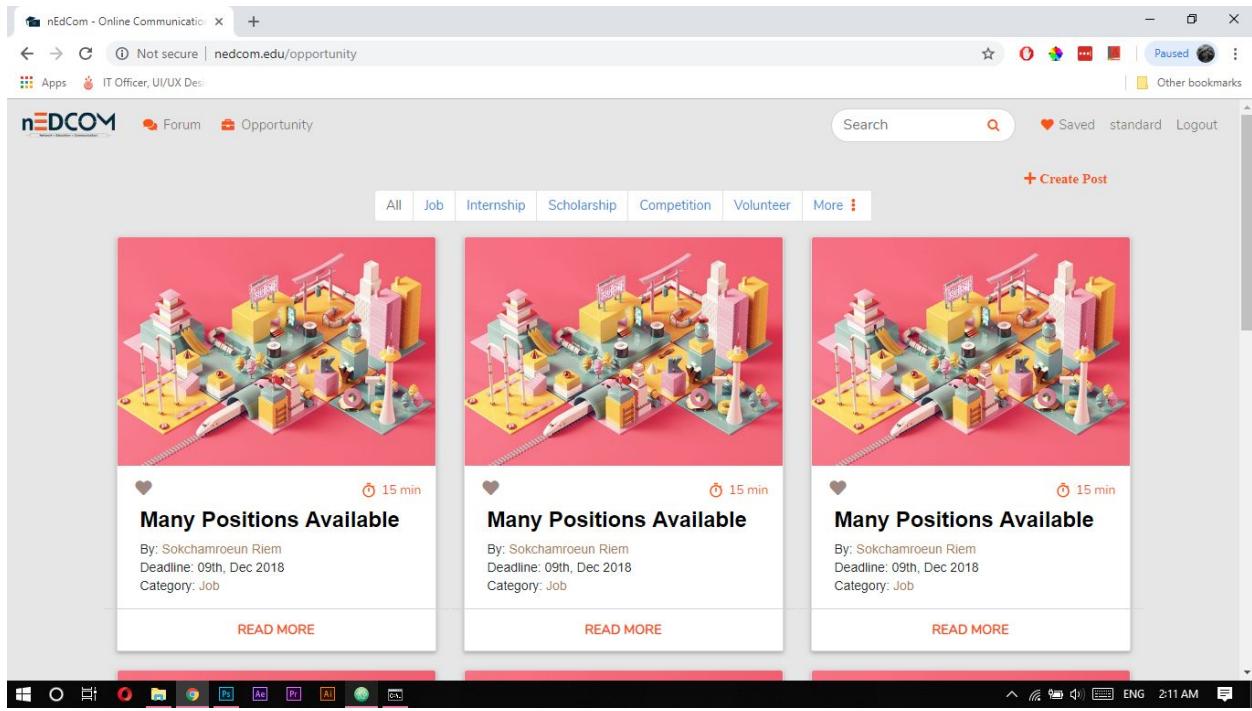


Figure 11.1.e: Opportunity homepage

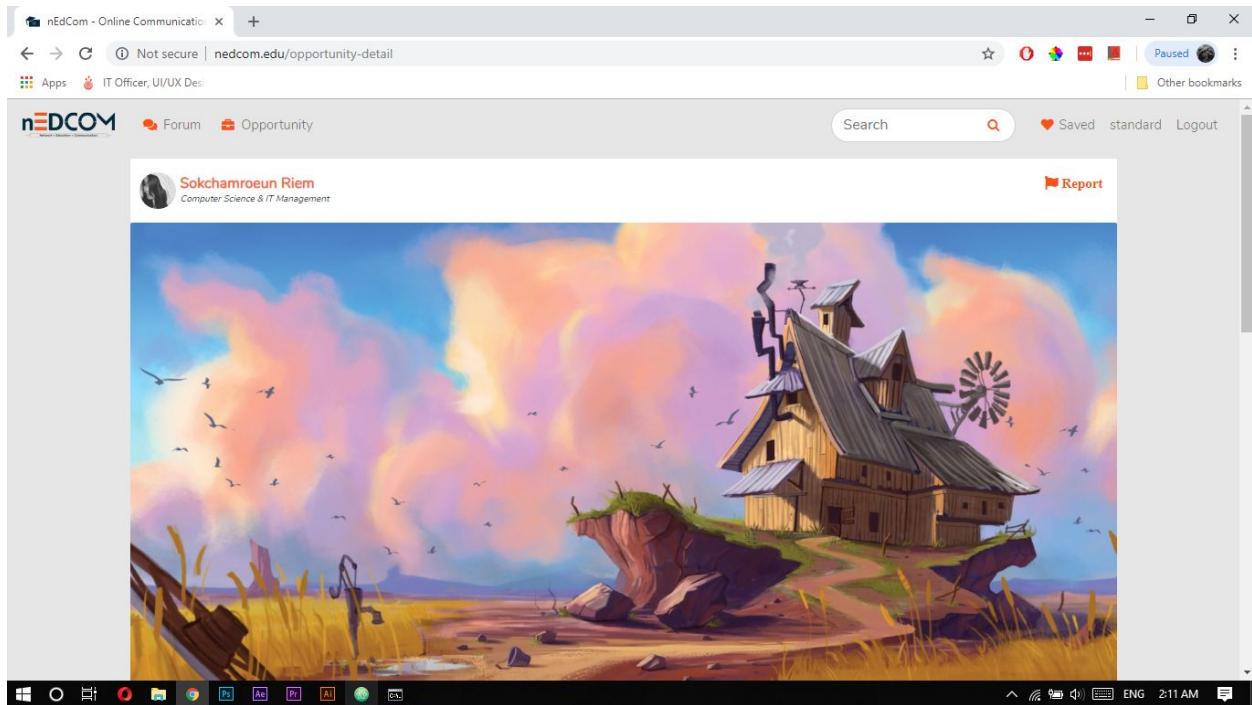


Figure 11.1.e: Opportunity Detail

The screenshot shows a web browser window for the nEdCom website. The URL is nedcom.edu/opportunity-detail. The page title is "Many Positions Available". It features a large image of a landscape with rocks and grass. Below the image, there is a section titled "Eligibility" with detailed requirements. Another section titled "Benefit" states that MCW Global will cover program and flight costs. The browser's toolbar and taskbar are visible at the top and bottom respectively.

Figure 11.1.e: Opportunity Detail

The screenshot shows a web browser window for the nEdCom website. The URL is nedcom.edu/forum. The page title is "Upload Image". A file named "47204276\_740093769684188\_7721014398227054592\_n.jpg" is selected for upload. The background shows a forum post with the title "Leave No One Behind: End Violence Against Women and Girls" and an orange graphic. The browser's toolbar and taskbar are visible at the top and bottom respectively.

Figure 11.1.e: Opportunity Create

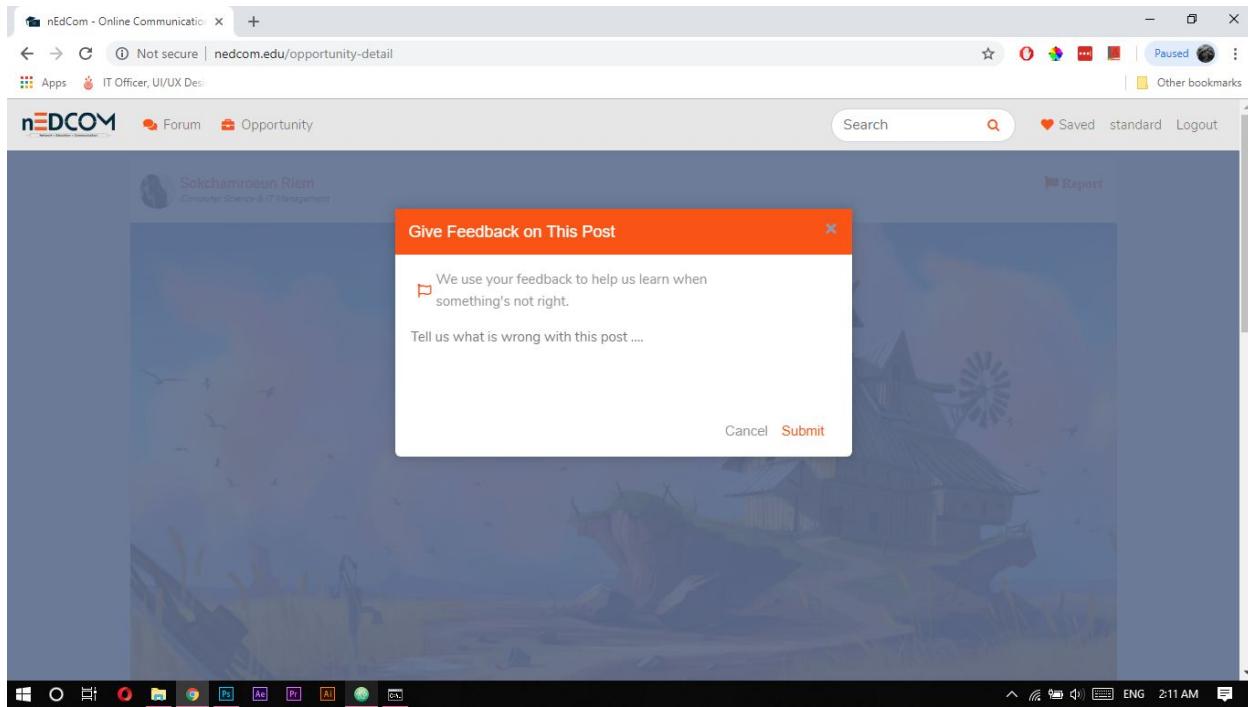


Figure 11.1.e: Opportunity Report

### 11.1.f. User Profile Page

The screenshot shows a web browser window for the nEdCom website. The URL is nedcom.edu/profile#. The page title is 'nEdCom - Online Communication'. The main content area displays a user profile for 'Sokchamroeun Riem'. The profile picture is a circular image of a person with dark hair. Below the picture, the user's name is 'Sokchamroeun Riem' and their bio is 'Computer Science & IT Management American University of Phnom Penh'. Below the bio are statistics: 130 Followers, 12 Following, 23 Questions, and 10 Answers. A 'Follow' button is visible. Below the profile section is a post titled 'What Is a Good GPA? What's a Bad GPA for College?' with 123 upvotes, 5 comments, and 150 views, posted 12 days ago. The post has a 'Report' link. The background of the page shows a collage of cartoonish animal faces.

Figure 11.1.f: User Profile Page

### 11.1.g. Admin Dashboard Page

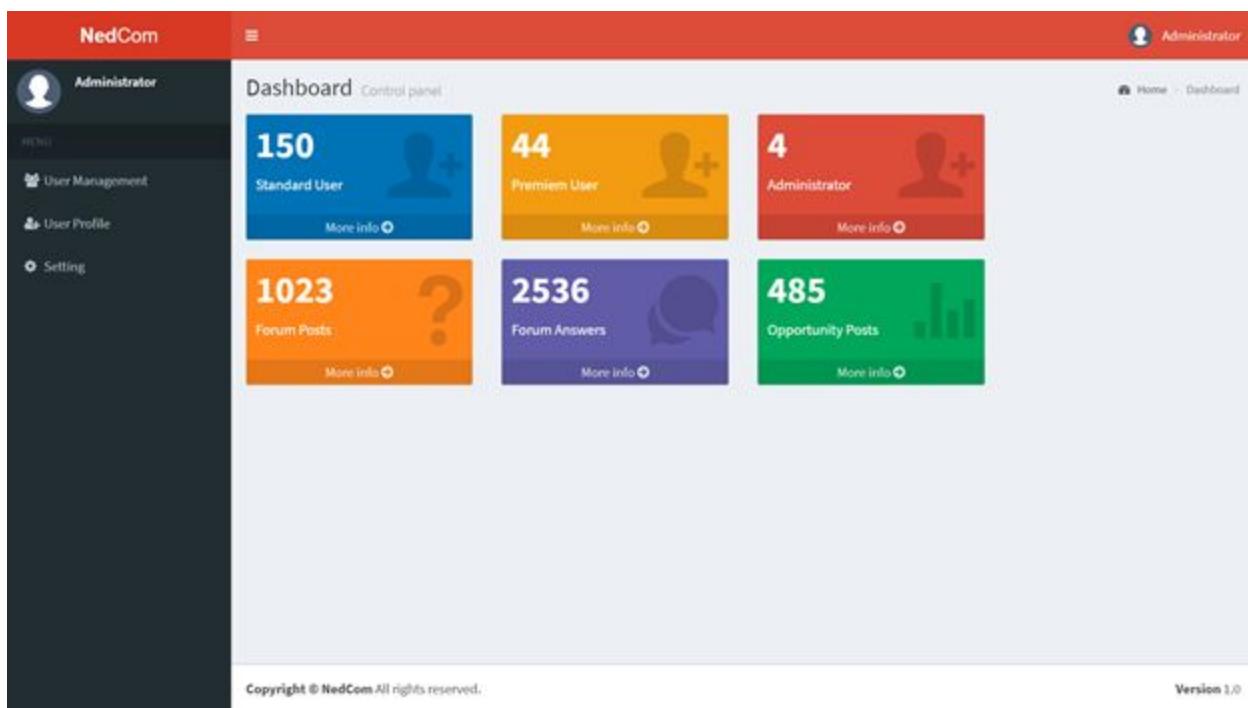


Figure 11.1.g: Admin Dashboard Page

#### 11.1.g. Admin User Management Page

User Management							
Account							
ID	Username	First Name	Last Name	Email	Registered Date	Status	Role
183	johndoe	John	Does	johndoe@gmail.com	11-7-2014	Approved	Standard
184	stonejohn	Stone	John	stonejohn@gmail.com	11-7-2014	Pending	Premium
185	brucewayn	Bruce	Wayne	bruceways@gmail.com	11-7-2014	Approved	Admin
183	johndoe	John	Does	johndoe@gmail.com	11-7-2014	Approved	Standard
184	stonejohn	Stone	John	stonejohn@gmail.com	11-7-2014	Pending	Premium
185	brucewayn	Bruce	Wayne	bruceways@gmail.com	11-7-2014	Approved	Admin
184	stonejohn	Stone	John	stonejohn@gmail.com	11-7-2014	Pending	Premium
183	johndoe	John	Does	johndoe@gmail.com	11-7-2014	Approved	Standard

Figure 11.1.g: Admin User Management Page

#### 11.1.h. Admin User Profile Management Page

The screenshot displays the NedCom Admin User Profile Management Page. On the left, a dark sidebar menu titled 'Administrator' includes 'User Management', 'User Profile', and 'Setting'. The main content area is titled 'User Profile' for 'John Doe' (Software Engineer). Key statistics shown are 132 Followers, 543 Following, 257 Questions, and 87 Answers. A red banner at the bottom indicates 'Points: 350'. The right side contains sections for 'About Me' (Bio: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam fermentum enim neque.'), 'Education' (M.S. in Computer Science from Fort Hays State University, B.S. in Computer Science from Fort Hays State University), 'Work Experiences' (Software Engineer at Microsoft Corporation, Android Developer at Google LLC), 'Volunteer Experiences' (Teaching Assistant at Fort Hays State University, Vice President of Student Government Association of FHSU, Event Organizer at CoderDojo), and 'Location' (Hays, Kansas, USA). The bottom of the page features a copyright notice 'Copyright © NedCom All rights reserved.' and a version number 'Version 1.0'.

Figure 11.1.h: Admin User Profile Management Page

## 12. Design of Tests

### Integration Testing

Following the test-driven development practice, tests are designed on the system to find bugs and identify them. For this system, integration testing strategy is used. Small units are tested first, with all the scenarios. Upon finish testing each module, integration test will be performed by combining all testing units into one system component to ensure all the smaller components work properly together.

### User Acceptance Test Cases

The following tests are created based on the use cases, which are part of the system requirements from report 1:

- Register Standard User
- Register Premium User
- Login
- Post in Forum
- Post in Opportunity Page
- User Edit profile
- Admin Manage User Account

**Test Case Identifier:** TC1-Register Standard User

**Pass/Fail Criteria:** The test passes if the user enters an email that is not existed in the database yet and has a correct email format, a first name, last name, password consisting of 6 or more characters

<p>and the confirm password matches</p> <p><b>Input Data:</b> First name, last name, email, password, retype of password</p>	
Test Procedure	Expected Result
Step 1. Enter valid email, first name, last name and incorrect confirm password  Step 2. Enter valid email, first name, last name and matching confirm password  Step 3. Click on the confirmation email	System alerts user of wrong confirm password and prompts user to retype the password  System sends confirmation email to user and notify user to click on the link in the email  System saves the account to the database, notifies user that account has been created successfully and directs user to homepage

<p><b>Test Case Identifier:</b> TC2-Register Premium User</p> <p><b>Pass/Fail Criteria:</b> The test passes if the user enters an email that is not existed in the database yet and has a correct email format, a first name, last name, institution, profession, specialty, password consisting of 6 or more characters, a matching confirm password, upload of proof of employment</p> <p><b>Input Data:</b> First name, last name, institution, profession, specialty, email, password, retype of password, upload of proof of employment</p>	
Test Procedure	Expected Result
Step 1. Enter valid email, first name, last name, institution, profession, specialty, and incorrect confirm password  Step 2. Enter valid email, first name, last name, institution, profession, specialty, and matching confirm password  Step 3. Clicks on the confirmation email  Step 4. Admin clicks approve user	System alerts user of wrong confirm password and prompts user to retype the password  System sends confirmation email to user and notify user to click on the link in the email  System saves the account to the database, notifies admin of premium account in need of approval, alerts user that an account has been created yet awaiting for admin approval and directs user to homepage  System update user account to premium in the database, alerts user that premium registration has been approved

<b>Test Case Identifier:</b> TC3-Login	
<b>Pass/Fail Criteria:</b> The test passes if user enters credentials contained in the database with less than number of allowed unsuccessful attempts.	
<b>Input Data:</b> Email, Password	
Test Procedure	Expected Result
Step 1. Enter valid email and incorrect password  Step 2. Enter valid email and matching password	System alerts user of authentication failure, record number of unsuccessful attempt and prompt user to retype login credential  System directs user to homepage

<b>Test Case Identifier:</b> TC4-View Forum	
<b>Pass/Fail Criteria:</b> The test passes if user clicks on the forum page and see list of all forum questions	
<b>Input Data:</b> Forum page link	
Test Procedure	Expected Result
Step 1. Click on the forum link	System display forum page to user with listing of all forum question

<b>Test Case Identifier:</b> TC5-Post In Forum	
<b>Pass/Fail Criteria:</b> The test passes if user fills in a forum question header, forum question detail, select one or more tags	
<b>Input Data:</b> Forum question header, forum question detail, question tag(s)	
Test Procedure	Expected Result
Step 1. Enter question header, question detail, but select no tag  Step 2. Enter question header, question detail, and select one tag	System alert user of unsuccessful question post and prompt user to select one or more tags  System saves the question to the database, notifies user that question has been created successfully and direct user to forum homepage

<b>Test Case Identifier:</b> TC6-Post In Opportunity Page	
<b>Pass/Fail Criteria:</b> The test passes if user fills in opportunity header, opportunity description, image, and selected a category	
<b>Input Data:</b> Opportunity header, opportunity description, image, category	
Test Procedure	Expected Result
Step 1. Enter header, description, select a	System alert user of wrong image format and

category, upload file of wrong format	prompts user to select again
Step 2. Enter header, description, select a category and upload image file with correct format	System saves the post to the database, notifies user of the post creation and direct user to opportunity homepage

<p><b>Test Case Identifier:</b> TC7-User Edit Profile</p> <p><b>Pass/Fail Criteria:</b> The test passes if user fills in profile information in the edit profile page such as name, university degree, employment position.</p> <p><b>Input Data:</b> First name, last name, major, education history, work experience, volunteer experience, profile picture</p>	
Test Procedure	Expected Result
Step 1. Select on edit button on the bio section of user profile	System directs user to edit bio page
Step 2. Enter first name, last name, major	System saves the information to the database and notifies user of information updated
Step 3. Select on the edit button on the Education section of user profile	System directs user to edit education information page
Step 4. Enters school name, degree, expected graduation year	System saves the information to the database and notifies user of information updated
Step 5. Select edit button on work experience section of user profile	System directs user to edit work experience page
Step 6. Enter workplace, position, number of years	System saves the information to the database and notifies user of information updated
Step 7. Select button to change profile picture	System opens pop up box for uploading a picture
Step 8. Upload photo that exceed max file size	System informs user of incorrect file upload
Step 9. Upload photo with correct format and appropriate file size	System saves the image to the database and display the new image on user profile

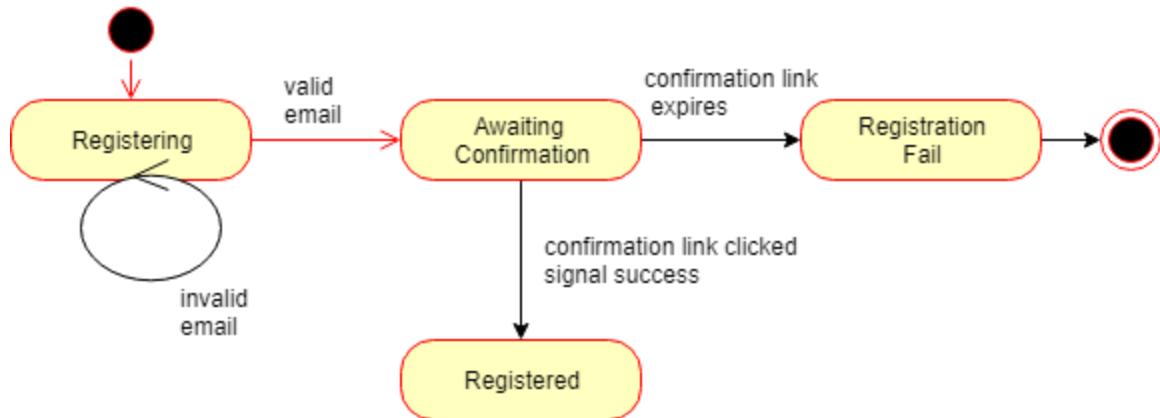
<p><b>Test Case Identifier:</b> TC8-Admin Manages User Account</p> <p><b>Pass/Fail Criteria:</b> The test passes if admin user clicks to view individual user profile information</p>
---

<b>Input Data:</b> User management links	
<b>Test Procedure</b>	<b>Expected Result</b>
Step 1. Login as admin role	System displays admin dashboard to the user
Step 2. Select User Management	System displays list of all user and their details such as name, email address, register date, status (approved/rejected/pending), and roles (admin/standard/premium)
Step 3. Select User Profile	System render user profile page that shows users bio, work, education, volunteer experience and user statistics (followers, following, questions, answers, point)

## State Diagram

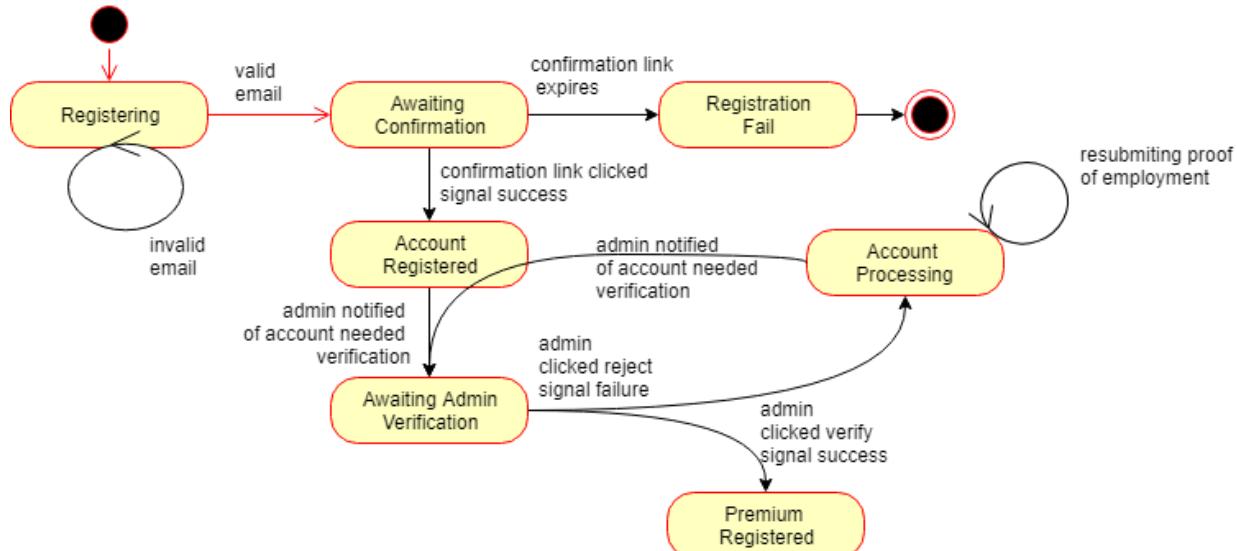
- Registering Standard Account

The RegisterController class will be tested. The Controller is responsible for verifying and creating all user role account. The user will need to enter required personal information and a valid email, then click on the confirmation link that they receive. If user enters an invalid email, they won't get to the next state of confirmation in progress, or if user fail to click on the link within a predetermined period, the account won't be created.



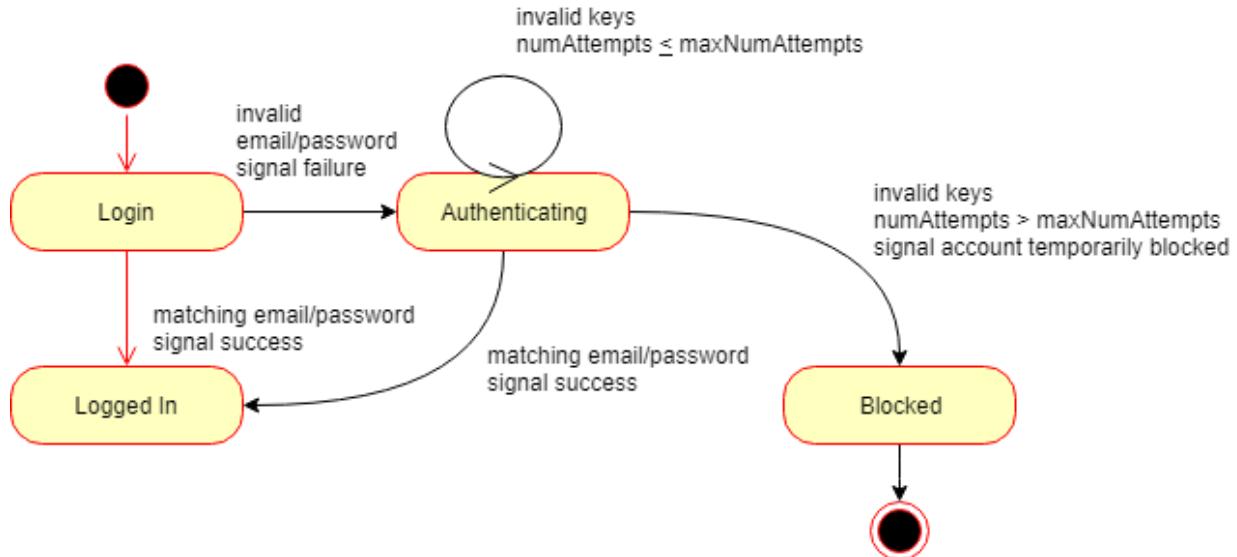
- Registering Premium Account

This test requires more steps in addition to the steps in the Registering Standard User test case above, since it needs the system admin to review and approve the request before the user account get registered as premium. Again, RegisterController will be tested. It should update user role to premium once the admin click on approve of premium registration.



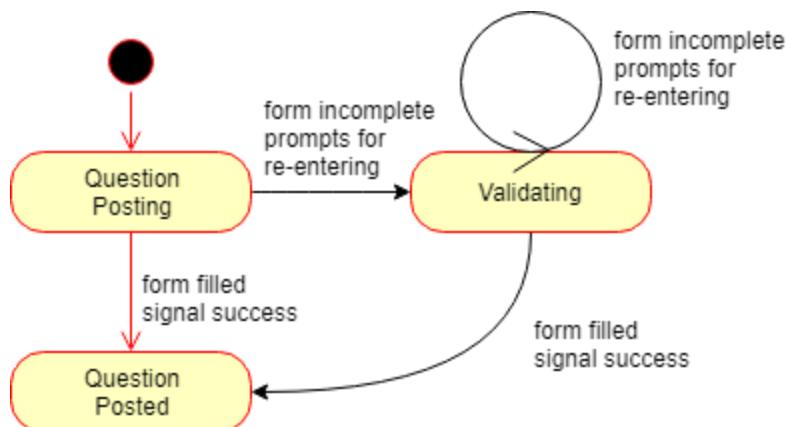
- Login

This test covers authentication of existing user account. LoginController class will be tested for verifying user while adding a layer of security by preventing user from entering wrong login information for a predetermined number of times. After series of unsuccessful attempts to login, the system will automatically block user from the system for a predetermined period of time.



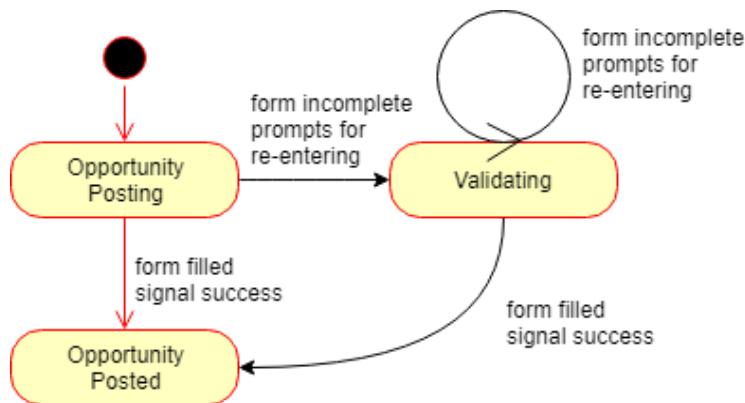
- Post Forum Question

ForumController class will be tested for the workings of forum. The Controller validate the form filled by user and prompts user to enter again if not all required fields are completed. The class saves the data into the database once it validates the data.



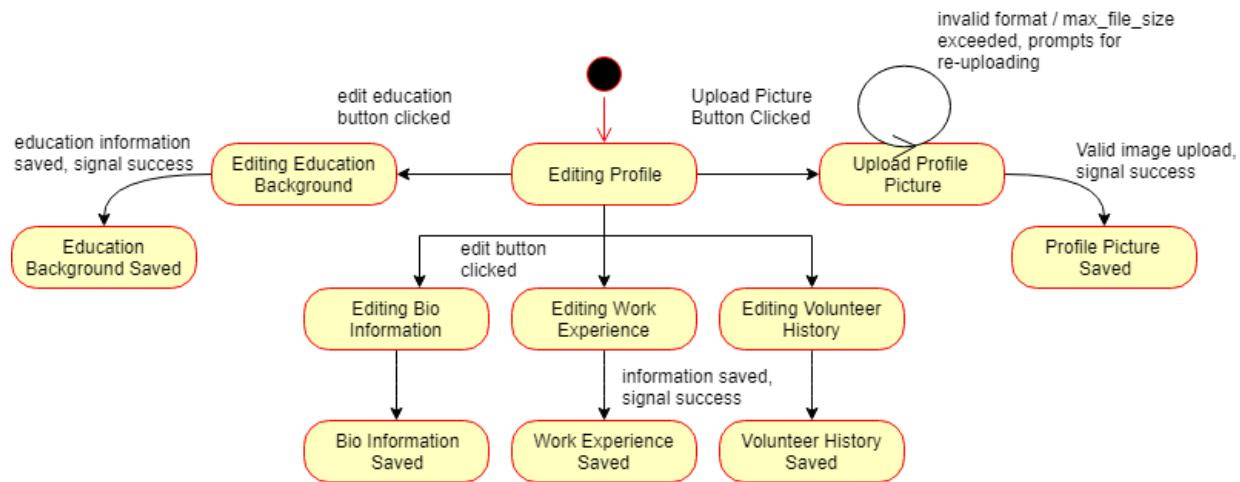
- Create Opportunity Post

In Creating Opportunity Post, OpportunityPagePostController will be tested. It receives data from user, validate that all required fields follows the required length and format, then the post shall save to the database. If user upload files that are too big or file that is not in the range of accepted format, user will need to select a new file.



- User Edit Profile

Test the UserController to obtain user profile information such as profile picture, bio, work, education, volunteer history and check that information is successfully written to the database. If user uploads profile picture that is too large or in wrong format, system will prompt user to re-upload.



## 13. Project Management

### a. History of Work, Current Status, and Future Work

- Molika Meas  
*Backend – Forum Management*
- Heanh Sok  
*Backend – User Management, Opportunity Page Management*
- Sokchamroeun Riem  
*Frontend – User Account, Forum, and Opportunity Page Interface*
- Kanika Montha  
*Frontend – User Profile, Forum Interface*

In the process of developing nEdCom all team members had put out equal contribution.

The team conducts weekly meetings to ensure update of work progress and responsibility division. Two members focus on front-end design, crafting logos and producing responsive, interactive web pages. In the meantime, the other two work on the backend components such as designing the computing logic, classes, functions, and the database structure. Finally, the front-end and back-end codes are integrated, and we run unit tests on each component, making sure that each individual module works fine, before testing the whole system. As we build the platform, we iteratively add, remove, and change priority of some functionalities, making the system progresses over time.

### \*\*\*\*\* Key Implementations \*\*\*\*\*

To serve the aforementioned purpose of nEdCom as an educational, networking, and communication platform for university students, the following functionalities have been implemented and tested:

#### 1. Login, Authentication, Signup, User Profile

These are fundamental for our system to work since we require restrictions on user access level to perform any actions such as posting and sharing contents on our website. On top of this, users can edit their profile information, upload profile picture, view their rating points, as well as view other user's profile.

#### 2. User Account Management (Admin Dashboard)

Admin dashboard for user account management is created for system administrator to monitor the number of users, questions and answers posted, opportunity posts shared, and useful data. In addition to the statistical information, administrator can manage user accounts such as view, edit, or delete user account from the system. It comes in handy when administrator needs to restrict users who violates the use of the system or reset account when user forgets login informations.

#### 3. Forum

Forum is one of our core features, which enables users to share information as a community through posting questions, submitting answers, voting on posts of other users and reading previously discussed topics. Users can filter question posts by tags, and select multiple tags for their question when posting.

#### 4. Opportunity Page

Opportunity Page is the collection of posts such as scholarships, job announcements, internship, and volunteering opportunities. Only users with premium access are allowed to post in the opportunity page, yet all types of users have access to the post, click to view the post detail, or bookmark posts for later view.

#### \*\*\*\*\* Possible Direction for Future Work \*\*\*\*\*

As nEdCom grows, it will extend to fit the need of a wide variety of students, employers, employees, and professionals who use the platform as the mean to sharing and networking. The following functionalities are saved for future implementation:

##### 1. Private chat

Allows users to directly inbox other users

##### 2. Mentoring Service

Enable users to apply to be a mentor and host private mentorship sessions, which other users can make booking requests and rate or review their experience after participating in the session.

#### b. References

- Problem Statement:
  - Reaksmeay, Hul, and VOA Khmer. “World Bank: More Skilled Worker Training Needed to Sustain Cambodia Growth.” *VOA*, VOA, 15 May 2018, [www.voacambodia.com/a/more-skilled-workers-training-needed-to-sustain-cambodia-growth-says-world-bank/4394843.html](http://www.voacambodia.com/a/more-skilled-workers-training-needed-to-sustain-cambodia-growth-says-world-bank/4394843.html).
  - “Three Keys to Unlocking Cambodia's Growth.” *East Asia Forum*, 12 Nov. 2014, [www.eastasiaforum.org/2014/11/13/three-keys-to-unlocking-cambodias-growth/](http://www.eastasiaforum.org/2014/11/13/three-keys-to-unlocking-cambodias-growth/).
- Report Guideline:
  - Marsic, Ivan. “Software Engineering Project Report.” *Book: Computer Networking - Textbook by Ivan Marsic*, www.ece.rutgers.edu/~marsic/Teaching/SE/report1.html.
- Course Book: Software Engineering by Ivan Marsic  
[http://www.ece.rutgers.edu/~marsic/books/SE/book-SE\\_marsic.pdf](http://www.ece.rutgers.edu/~marsic/books/SE/book-SE_marsic.pdf)