Rita Samir
ID : 25

Sohayla Mohammed
ID : 31

Merit Vector
ID : 69

Youmna Dwidar
ID : 73

CSED GP Library: Milestone 3

27th April 2020

OVERVIEW

We proposed a web application represented as a library of resources for all previous graduation projects and for the upcoming graduation projects, students can upload their codes and their documentation. We also propose the web application as an open environment for communication and sharing experience between all computer science and engineering students.

In this milestone we started the development of the second Agile increment.

Requirements Elicitation

I.   Requirements Specification

We used Structured specifications to illustrate our requirements :

Function	Creation of new project post
Inputs	Post information : Description/Abstraction, Field(s), User Citations(student, Teaching staff (TS)), Paper/document.
Outputs	A message to the user that his post will be reviewed and response will be sent to him shortly
Action	After validating all post information, adding this post to the pending posts list for teaching staff to verify.
Requirements	User is authenticated and already logged in the site.
Pre-conditions	- All requirement fields are not empty. - All post information is valid.
Post-conditions	Adding post to pending posts list waiting for teaching staff members to approve or disapprove it.
Side effects	None

Requirement validation by reviews :

Review checks :

Verifiability	Comprehensibility	Traceability	Adaptability
✔	✔	✔	✔

Function	Verifying a project post
Inputs	List of the posts to be verified each post has : Description/Abstraction, Field(s), User Citations(student, Teaching staff (TS)), Paper/document.
Outputs	Decision of the teaching staff (approve/disapprove) sent as a notification to the post creator and all the post's citations.
Action	Show pending posts to the teaching staff and take their response about the posts and send it to the creator and the citations and do the appropriate action to the post depending on the teaching staff response.
Requirements	- User is authenticated and already logged in the site. - User is logged in the site as a teaching staff member.
Post-conditions	- if the post is approved, adding this post to the posts list to be shown in the home page. - If the post is disapproved, remove this post after sending the rejection to the creator and the citations.
Side effects	None

Requirement validation by reviews :

Review checks :

Verifiability	Comprehensibility	Traceability	Adaptability
✔	✔	✔	✔

Function	Adding a comment to a project post
Inputs	The comment body, the user who is writing the comment and the post which the comment attached to.
Outputs	Notification sent to all the post's citations and followers (next milestone)
Action	After the user clicks the add comment button, the comment will be added to the database with its relations to post and to the user. The comment will be shown in the home page under its post and every one can see the comment.
Requirements	User is authenticated and already logged in the site.
Post-conditions	Adding the comment to the database.
Side effects	None

Requirement validation by reviews :

Review checks :

Verifiability	Comprehensibility	Traceability	Adaptability
✔	✔	✔	✔

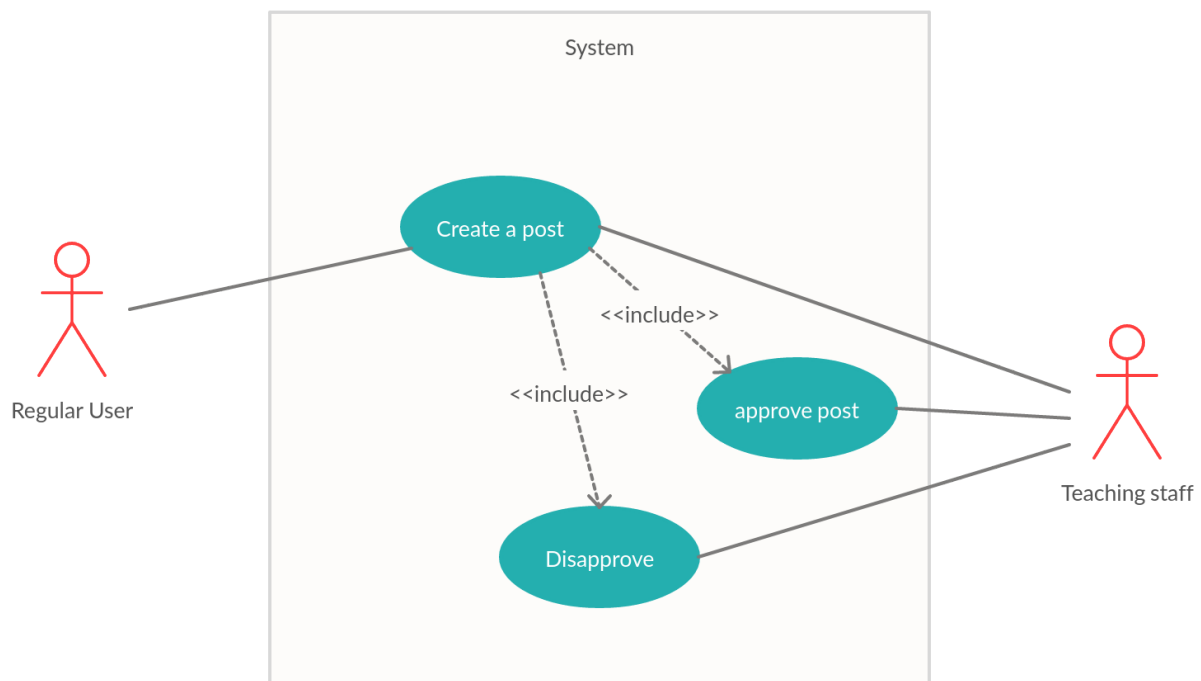| Function	Search for a project post
Inputs	Key search : - title or abstract of the post - Name of the citation users of the post
Outputs	All posts that might have the key search.
Action	User enters the key search that he is looking for, searching the database for all posts that match the key search and display all the posts to the user.
Requirements	No requirement required any type of user can do this function
Pre-conditions	None
Post-conditions	Display all posts on the home page.
Side effects	None

Requirement validation by reviews :

Review checks :

Verifiability	Comprehensibility	Traceability	Adaptability
✔	✔	✔	✔

Requirements validation by test-case generation :

We did requirement validations by creating test cases containing those requirements and running those tests on the current model.

II.   Use case diagrams
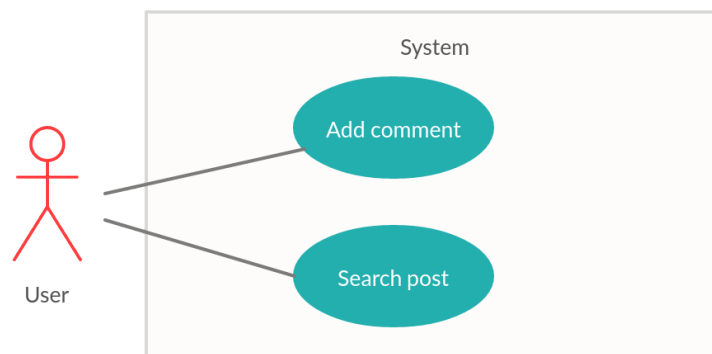
III. Activity diagrams

A. Sign in/up

B. Creation of a new project post

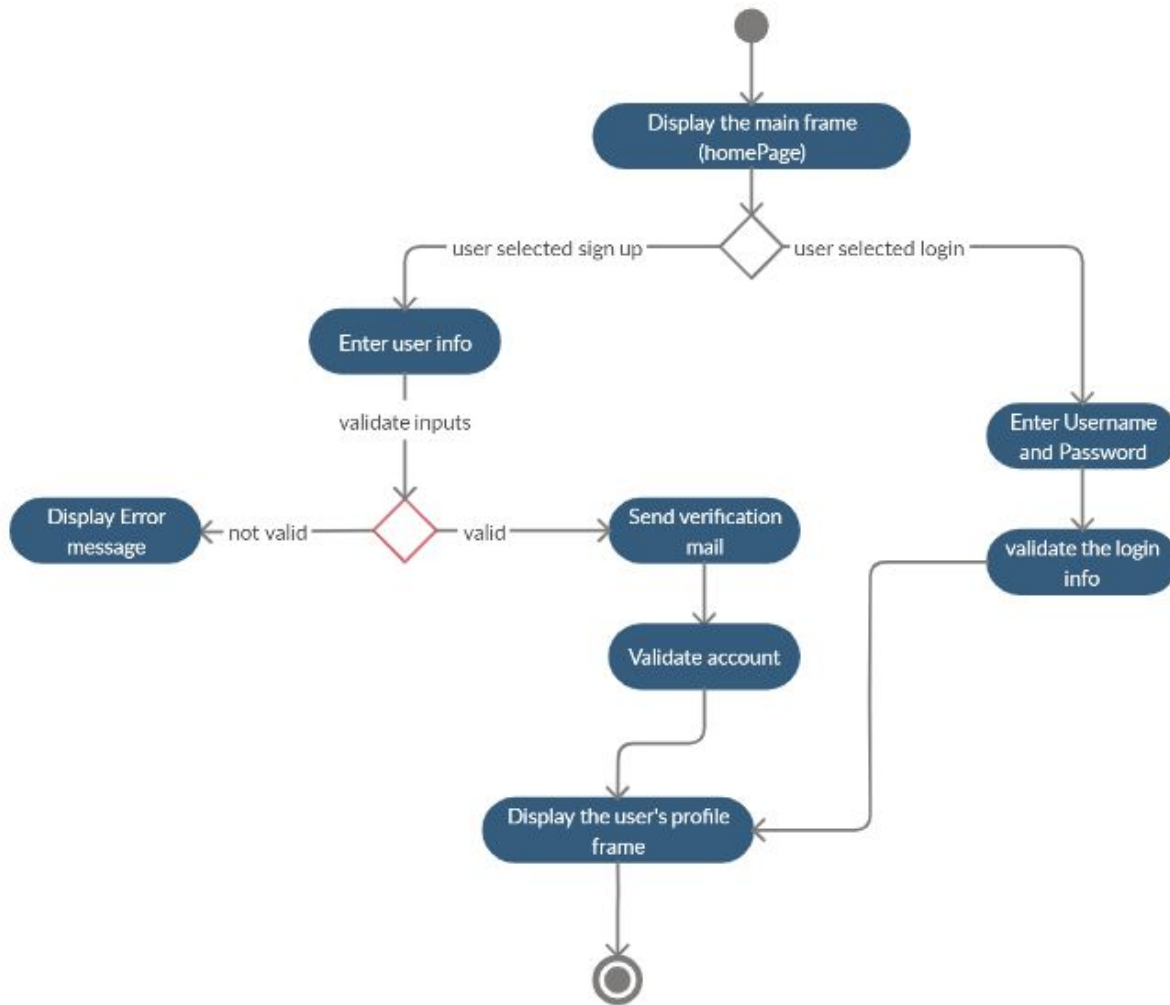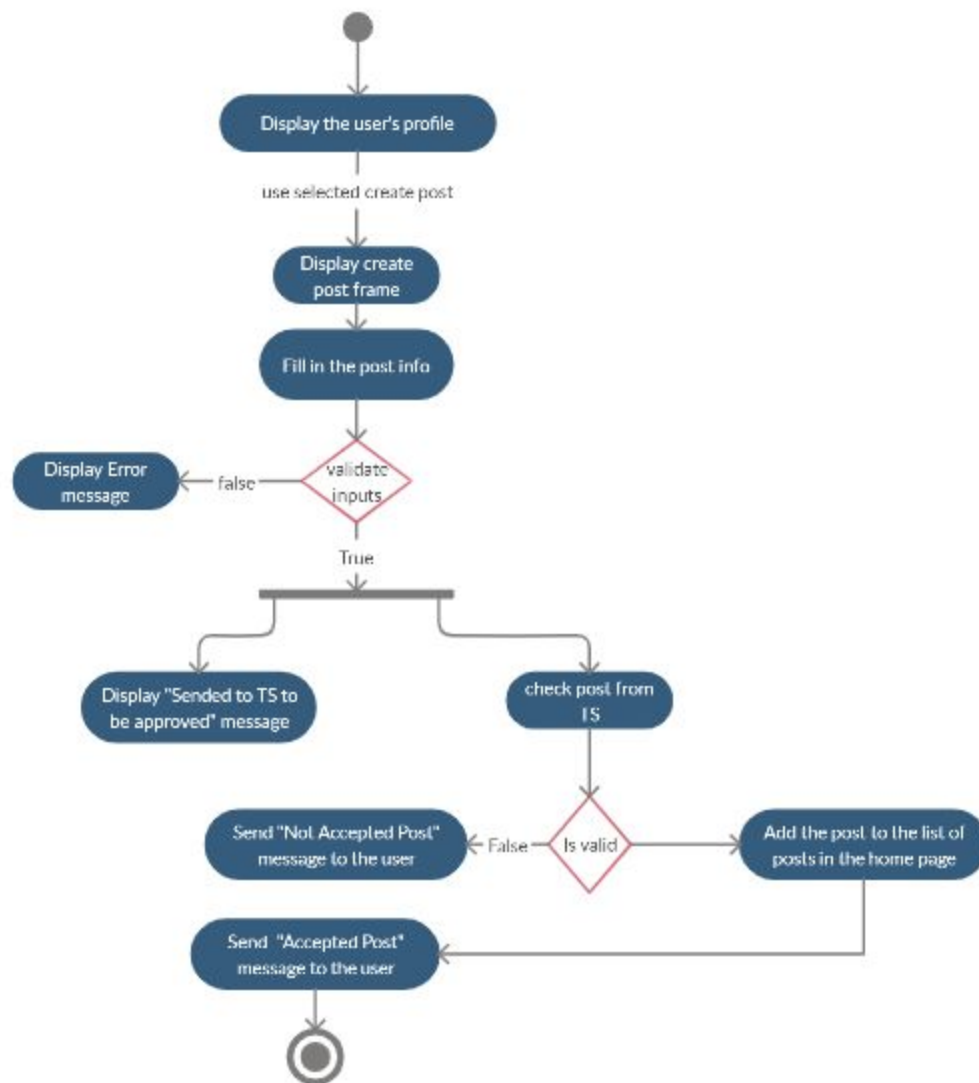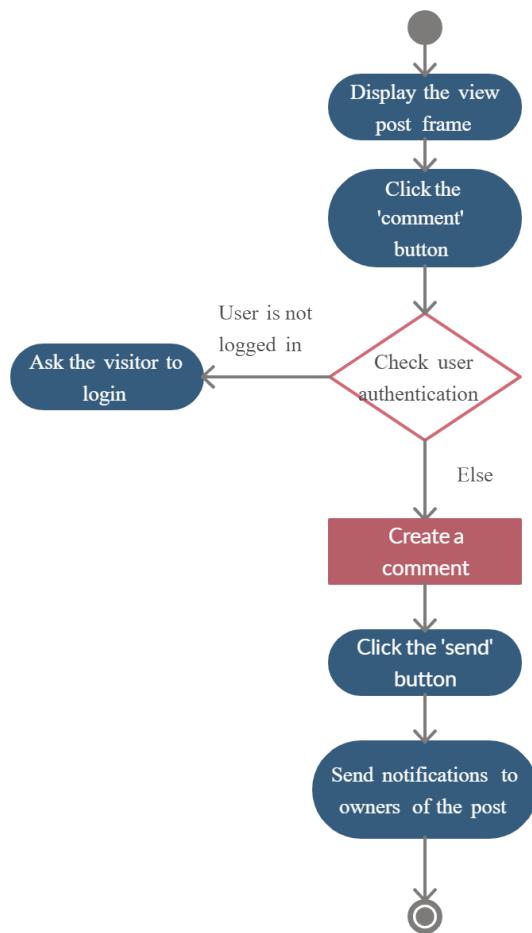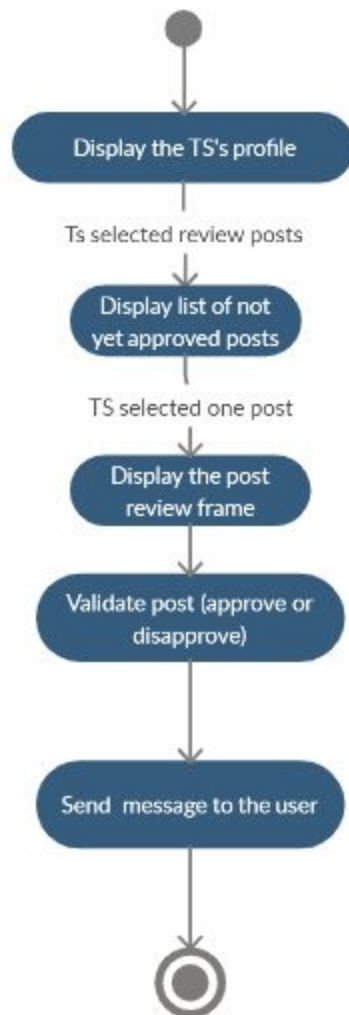## C. Add a comment to a post

```
                              ●

                    ┌──────────────────┐
                    │ Display the view │
                    │   post frame     │
                    └──────────────────┘
                              │
                    ┌──────────────────┐
                    │    Click the     │
                    │   'comment'      │
                    │     button       │
                    └──────────────────┘
                              │
      User is not             ◇
      logged in        Check user
┌──────────────┐      authentication
│ Ask the visitor to │◄──────
│     login     │
└──────────────┘          │ Else
                    ┌──────────────────┐
                    │    Create a      │
                    │    comment       │
                    └──────────────────┘
                              │
                    ┌──────────────────┐
                    │ Click the 'send' │
                    │     button       │
                    └──────────────────┘
                              │
                    ┌──────────────────┐
                    │ Send notifications to │
                    │ owners of the post │
                    └──────────────────┘
                              │
                              ◎
```

D. Teaching Staff (TS) verifying a project post

Display the TS's profile

Ts selected review posts

Display list of not yet approved posts

TS selected one post

Display the post review frame

Validate post (approve or disapprove)

Send message to the user

E. Search for a post

Display the main frame

Select category to search in

Type the search keywords and press 'search'

Found results?

No → Display 'could not find results' message

Yes → Display results preview frame

User selects one of the result

Display the view post frame

No more operations

IV.   Test cases

Tests for verifying posts:

	Functionality	Input	Output
Visitors can't add posts	Ensure that visitors can't add posts	Get request to creating a new post	Redirecting to login page
Logged in can add post	Allow logged in to add posts	Get request to creating a new post	Assert that the add post form is visible.
Fields are required	Ensure that fields field us entered by the user.	Form request without fields attribute.	Assert error and redirection to form
Title is required	Ensure that fields field us entered by the user.	Form request without fields attribute.	Assert error and redirection to form
Abstract is required	Ensure that Abstract field is entered by the user.	Form request without abstract attribute.	Assert error and redirection to form
Doc url is required	Ensure that doc url field is entered by the user.	Form request without doc url attribute.	Assert error and redirection to form
At least one TS	Ensure that at least one TS.	Form request from student without collaborators attribute.	Assert error and redirection to form
Student valid input	Ensure that the post is added correctly and that it's not approved yet.	Form request from student.	Assert that the records are added correctly and that it's not approved.
TS valid input	Ensure that the post is added correctly and that it's approved.	Form request from TS.	Assert that the records are added correctly and that it's approved.

Tests for verifying comments:

	Functionality	Input	Output
Visitors can't add comments	Ensure that visitors can't add comments	Get request to creating a new comment	Redirecting to login page
Logged in can add comments	Allow logged in to add comments	Get request to creating a new comment	Assert that the add comment form is visible.
body is required	Ensure that body field is entered by the user.	Form request without body attribute.	Assert error and redirection to form