flarem plus.

PROJECT PLAN DOCUMENT

Forumium: A Web Based Forum Platform for University Student

Objective

Project Date

Start: 11/11/2025

End: 9/1/2026

By the end of January 2026, we aim to release the first beta version of the forum platform to students for X University by hiring a team of 5 software developers with a budget of RM 150,000.

Technical Result

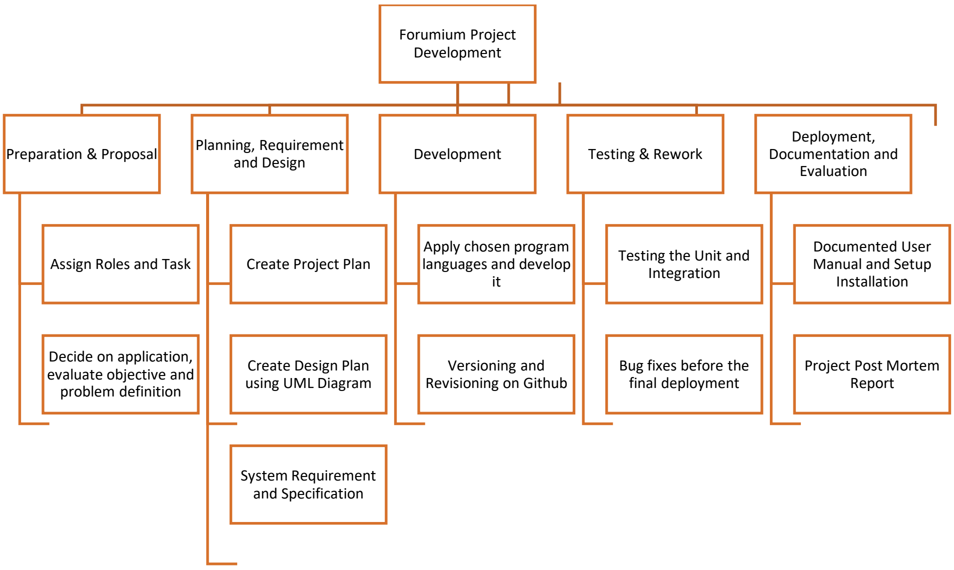
What to be produced?

Produce a working forum platform with all the requirement following the System Requirement Specification SRS

Specification of final product:

- Login and Logout System
- Change Account Password
- Create Post and Topics
- Search Topics
- Add Comments
- Reply Topics

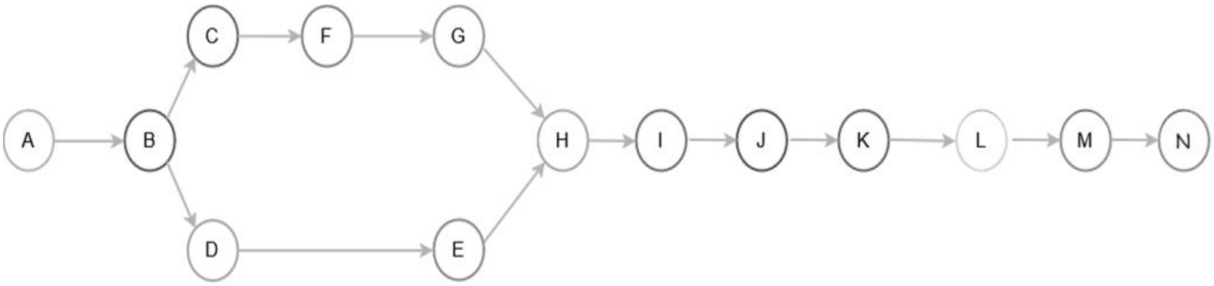
Work Breakdown Structure Diagram


Forumium Project Development
Preparation & Proposal
Planning, Requirement and Design
Development
Testing & Rework
Deployment, Documentation and Evaluation
Assign Roles and Task
Decide on application, evaluate objective and problem definition
Create Project Plan
Create Design Plan using UML Diagram
System Requirement and Specification
Apply chosen program languages and develop it
Versioning and Revisioning on Github
Testing the Unit and Integration
Bug fixes before the final deployment
Documented User Manual and Setup Installation
Project Post Mortem Report

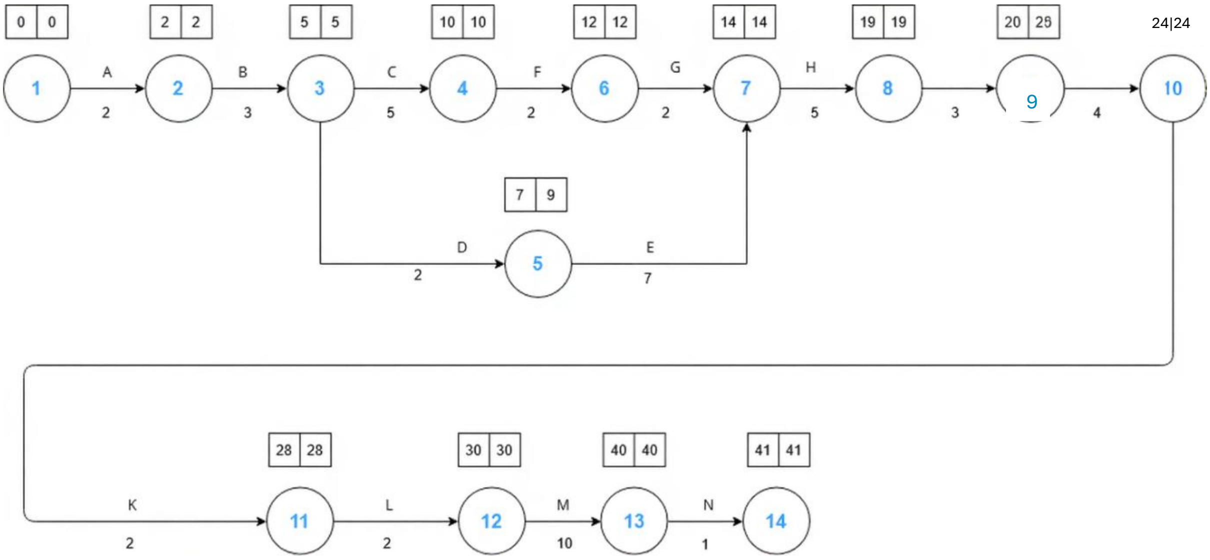
Activity Table

No.	Activity Description	Activity	Immediate Predecessor	Activity Time(days)
1	**Forum Platform Management and Preparation**			**5**
1.1	Gather Requirement	A	-	2
1.2	Project Proposal	B	A	3
2	**Platform Planning and Design**			**5**
2.1	Establish the selected requirement and design	C	B	5
3	**Platform Development**			**11**
3.1	Select Technical Framework based on the chosen language	D	B	2
3.2	Coding and Debugging	E	D	7
3.3	Create Content and Post	F	C	2
4	**Platform Testing and Rework**			**7**
4.1	Load Forum Content and Integration	G	F	2
4.2	Test site	H	E, G	5
5	**Platform Finalization**			**7**
5.1	Feedback form end user	I	H	3
5.2	Debug	J	I	4
6	**Platform Deployment**			**2**
6.1	Roll Out and Documentation	K	J	2
7	**Agile Sprint**			**12**
7.1	Creat Backlog from Feedback	L	K	2
7.2	Implement Changes	M	L	10
8	**Platform Launch**			**1**
8.1	Launch Forum Platform	N	M	1

AON Activity on Node



AOA Activity on Arrow



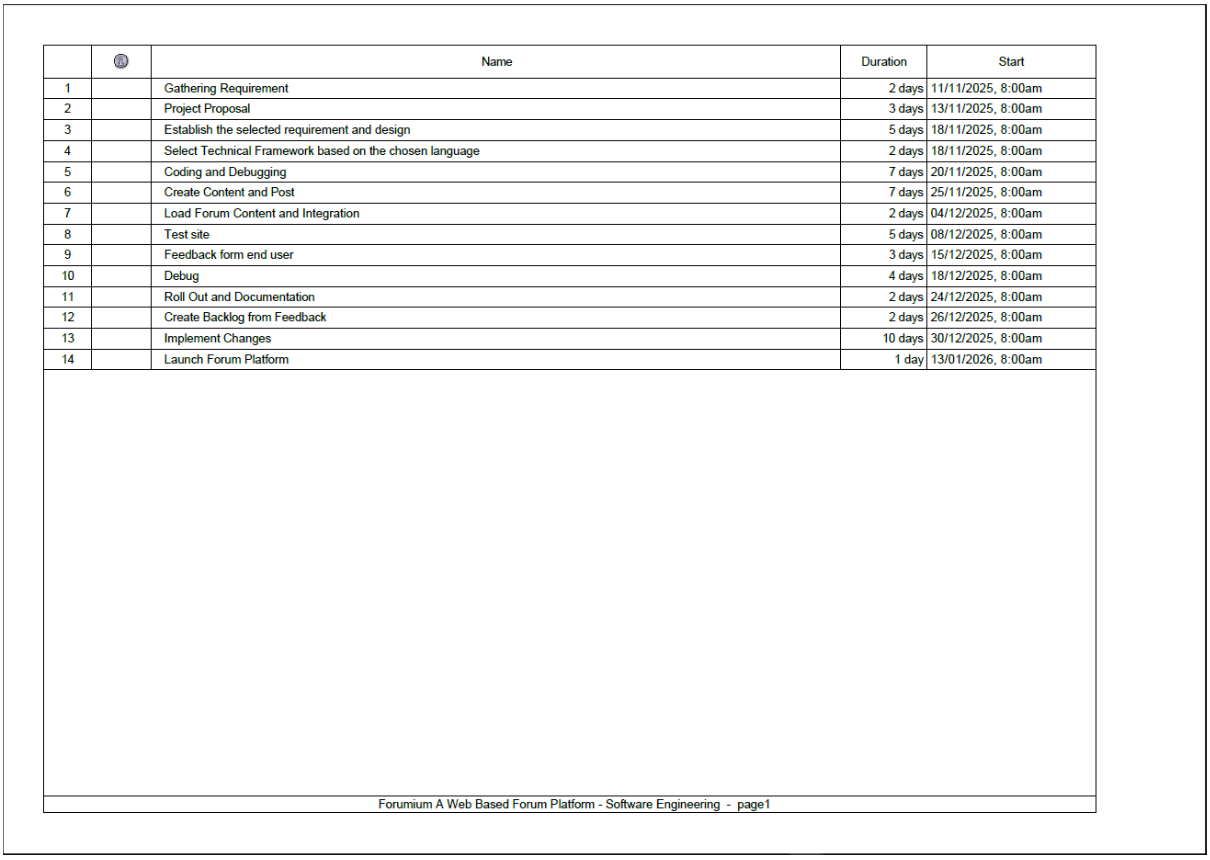
Crictical Path: A-B-C-F-G-H-I-J-K-L-M-N

Milestones

Milestone	Description	Responsible Personnel
Project Start	Define scope, objectives, SDLC model and assign roles	Project Manager
Requirements Completion	Gather requirements, confirm features, finalize tech stack	Lead Designer + Lead Programmer
System Environment Setup	Install server, configure database, install Flarum	Lead Programmer + Entire Team
Core Development	Customize UI, implement categories, install extensions	Lead Designer + Entire Team

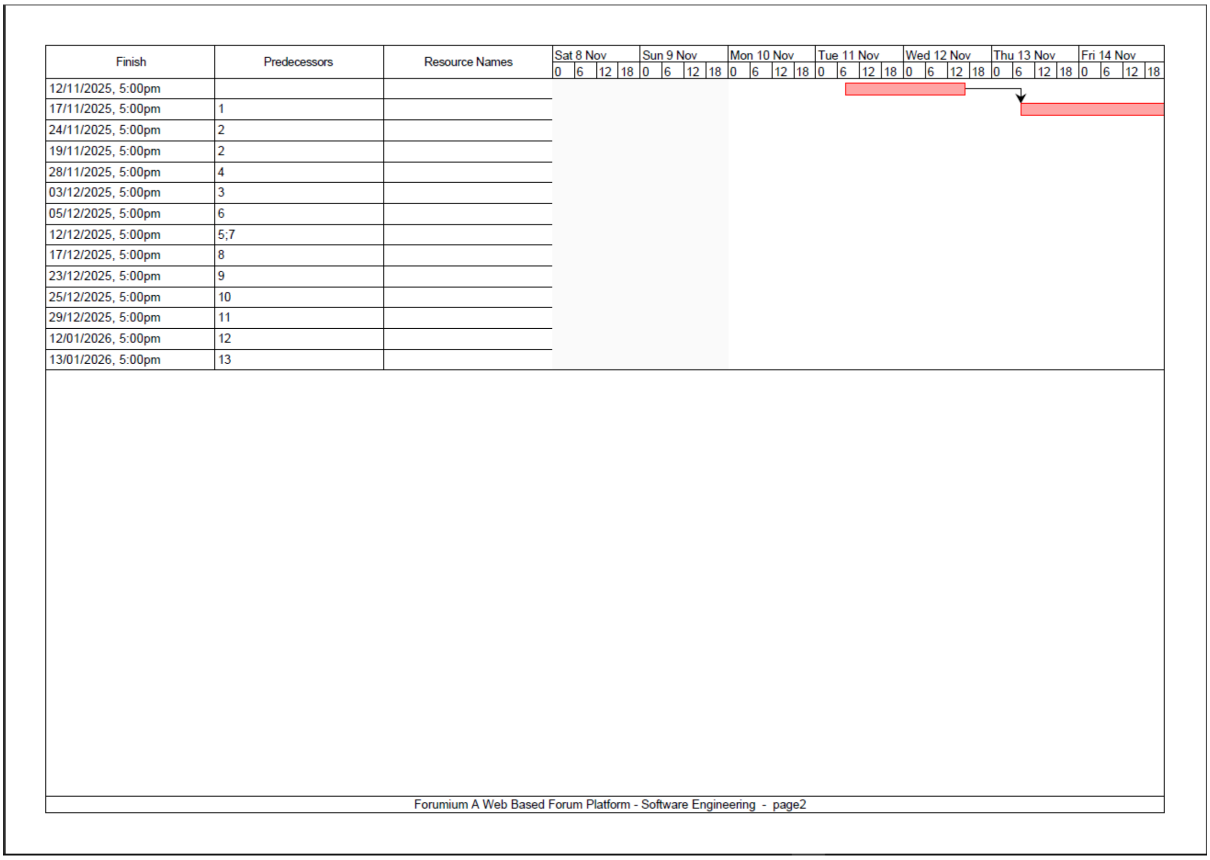
Testing Phase	Perform testing, record bugs and do rework	SQA + SCM
Deployment	Deploy forum, finalize server setup	Lead Programmer + Lead Designer
Final Documentation & Presentation	Prepare report, user guide, and slide deck	Documetation Manager + Entire Team

Gantt Chart



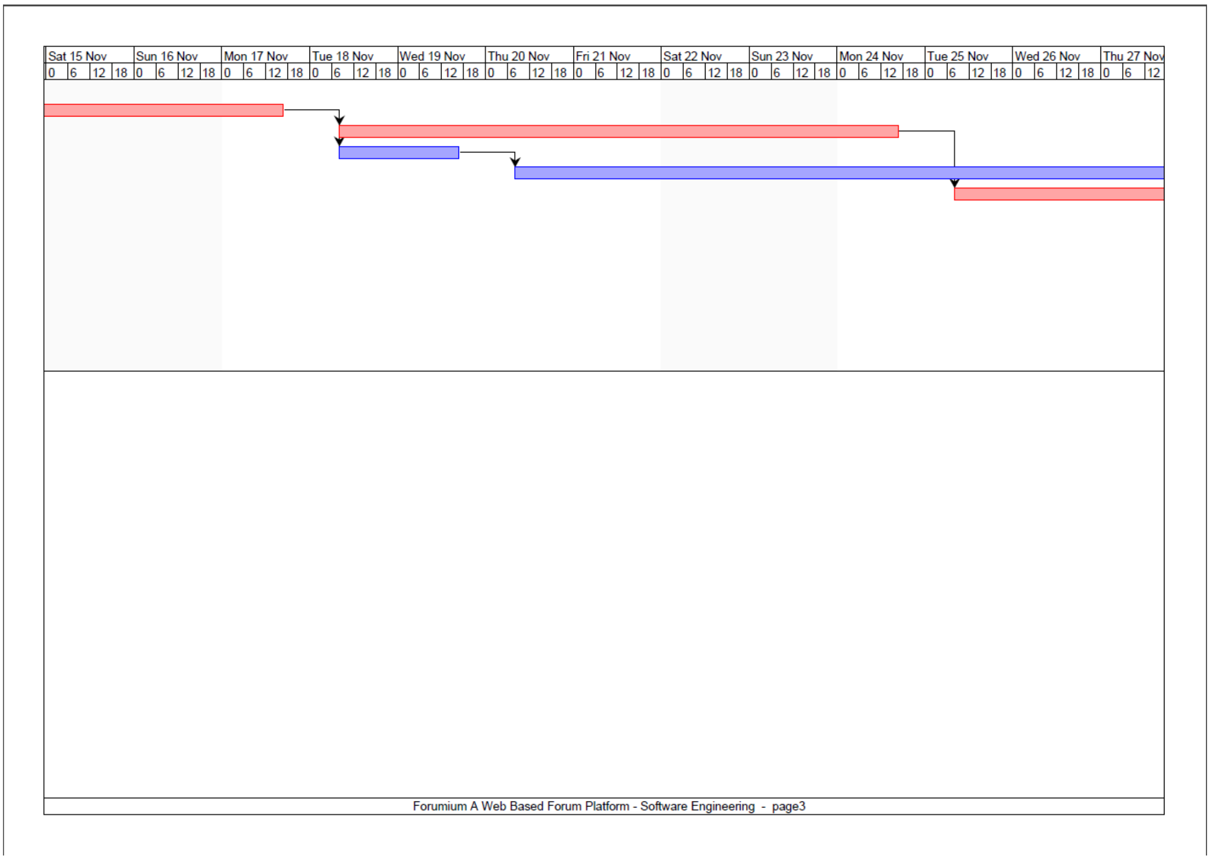
	ⓘ	Name	Duration	Start
1		Gathering Requirement	2 days	11/11/2025, 8:00am
2		Project Proposal	3 days	13/11/2025, 8:00am
3		Establish the selected requirement and design	5 days	18/11/2025, 8:00am
4		Select Technical Framework based on the chosen language	2 days	18/11/2025, 8:00am
5		Coding and Debugging	7 days	20/11/2025, 8:00am
6		Create Content and Post	7 days	25/11/2025, 8:00am
7		Load Forum Content and Integration	2 days	04/12/2025, 8:00am
8		Test site	5 days	08/12/2025, 8:00am
9		Feedback form end user	3 days	15/12/2025, 8:00am
10		Debug	4 days	18/12/2025, 8:00am
11		Roll Out and Documentation	2 days	24/12/2025, 8:00am
12		Create Backlog from Feedback	2 days	26/12/2025, 8:00am
13		Implement Changes	10 days	30/12/2025, 8:00am
14		Launch Forum Platform	1 day	13/01/2026, 8:00am

Forumium A Web Based Forum Platform - Software Engineering - page1

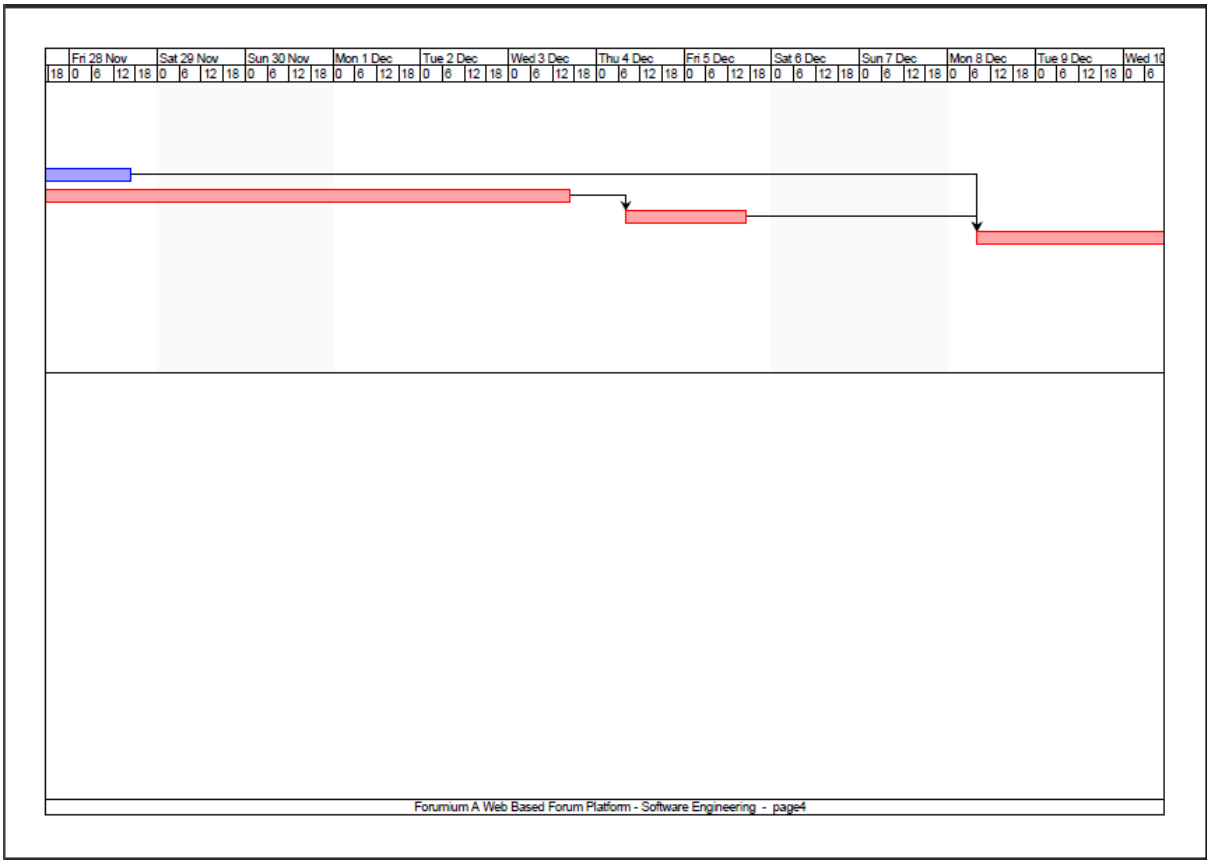


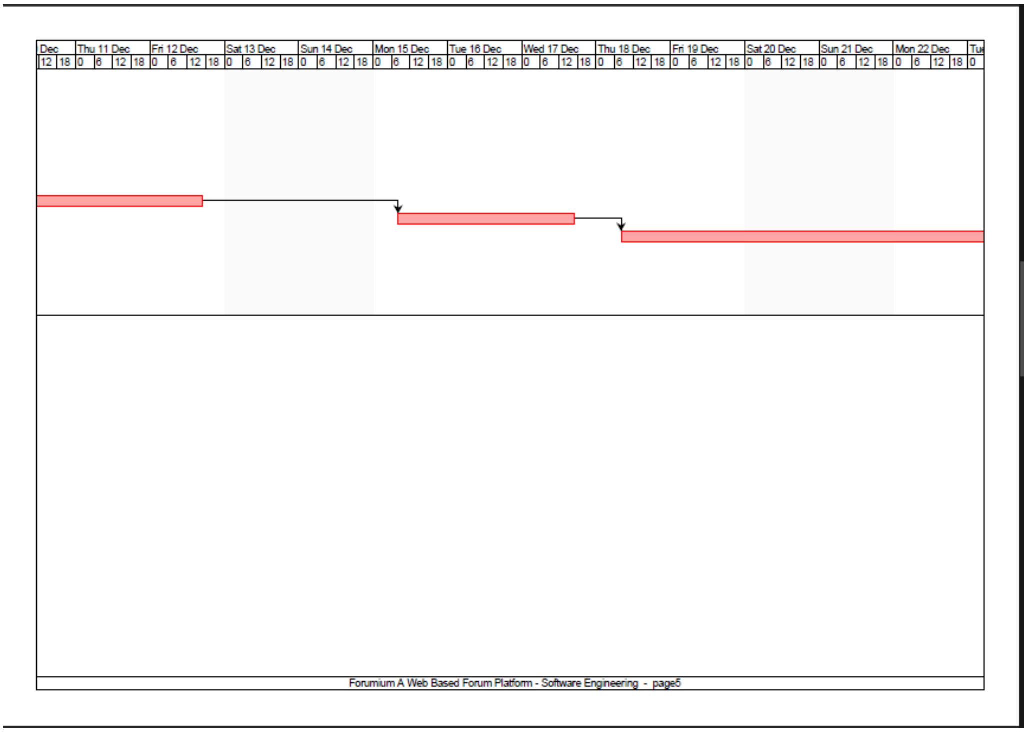
Finish	Predecessors	Resource Names
12/11/2025, 5:00pm		
17/11/2025, 5:00pm	1	
24/11/2025, 5:00pm	2	
19/11/2025, 5:00pm	2	
28/11/2025, 5:00pm	4	
03/12/2025, 5:00pm	3	
05/12/2025, 5:00pm	6	
12/12/2025, 5:00pm	5;7	
17/12/2025, 5:00pm	8	
23/12/2025, 5:00pm	9	
25/12/2025, 5:00pm	10	
29/12/2025, 5:00pm	11	
12/01/2026, 5:00pm	12	
13/01/2026, 5:00pm	13	

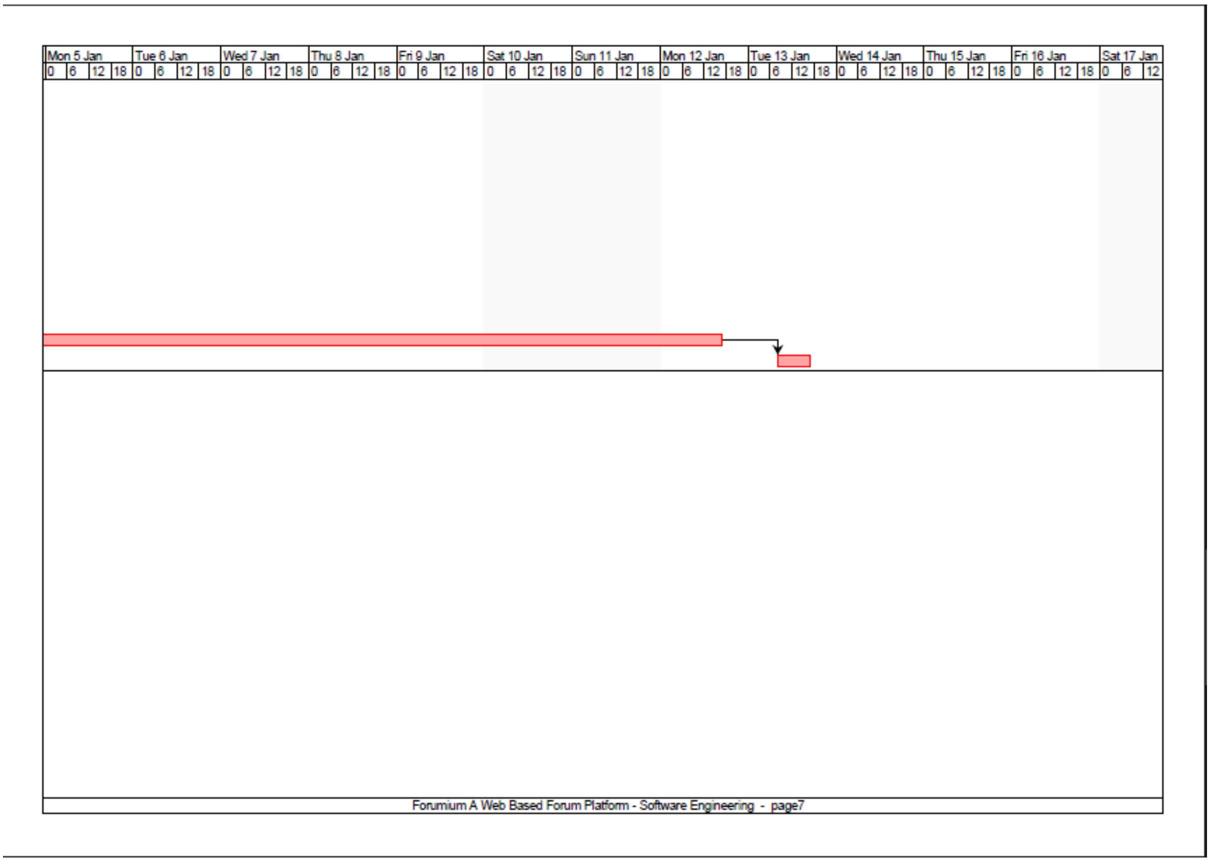
Forumium A Web Based Forum Platform - Software Engineering - page2

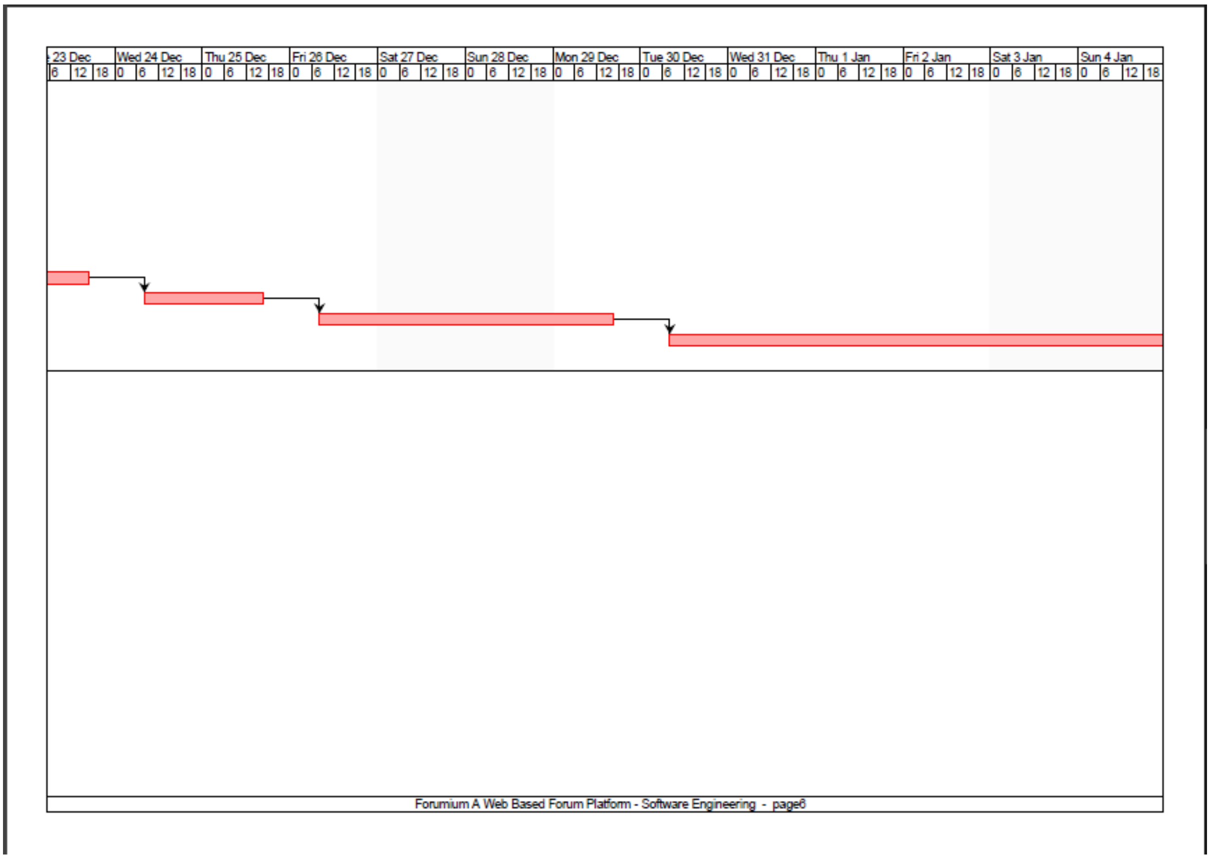


Forumium A Web Based Forum Platform - Software Engineering - page3

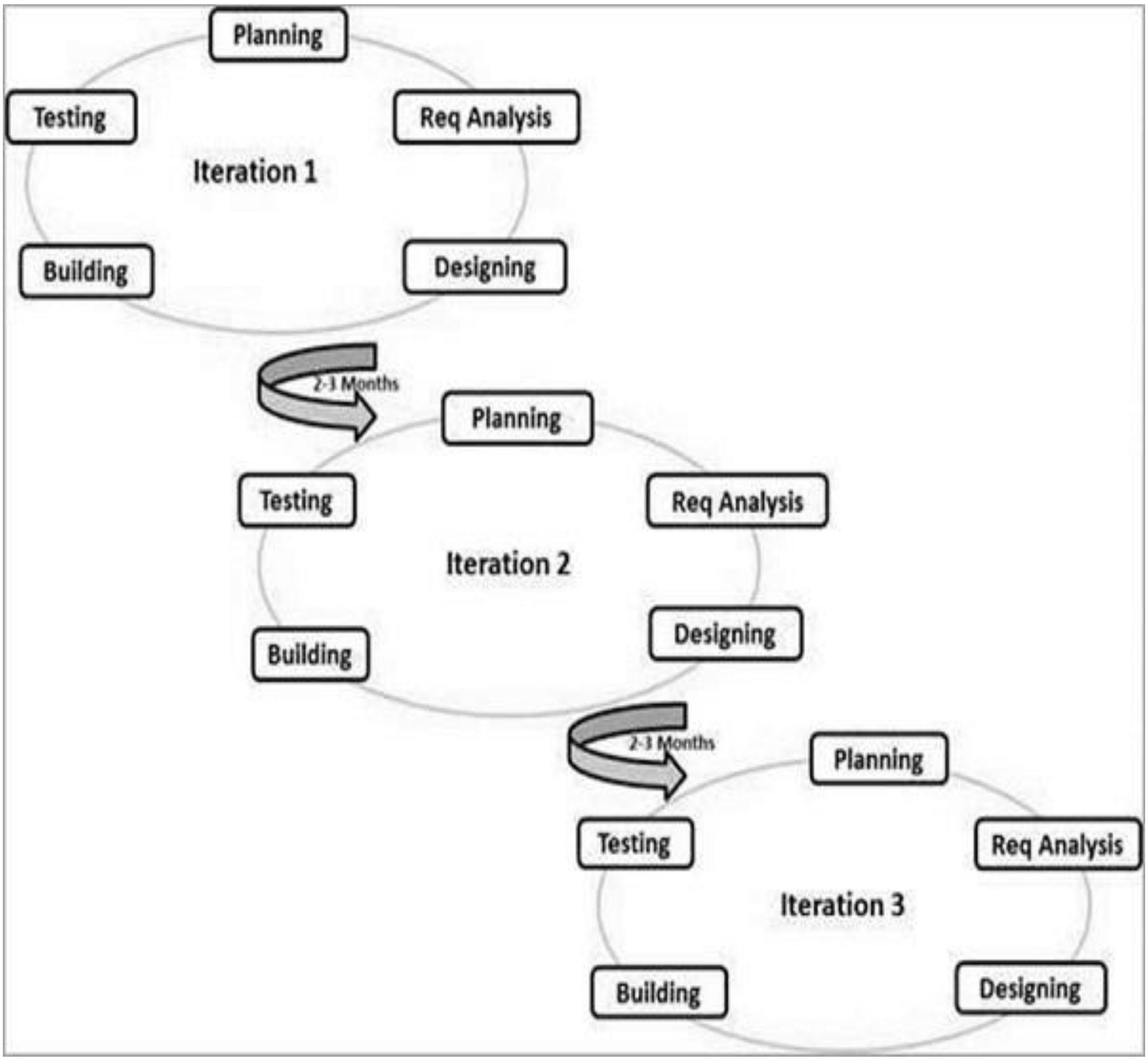

Fri 28 Nov
Sat 29 Nov
Sun 30 Nov
Mon 1 Dec
Tue 2 Dec
Wed 3 Dec
Thu 4 Dec
Fri 5 Dec
Sat 6 Dec
Sun 7 Dec
Mon 8 Dec
Tue 9 Dec
Wed 10
Forumium A Web Based Forum Platform - Software Engineering - page4


Dec
Thu 11 Dec
Fri 12 Dec
Sat 13 Dec
Sun 14 Dec
Mon 15 Dec
Tue 16 Dec
Wed 17 Dec
Thu 18 Dec
Fri 19 Dec
Sat 20 Dec
Sun 21 Dec
Mon 22 Dec
Tu
Forumium A Web Based Forum Platform - Software Engineering - page5


Mon 5 Jan
Tue 6 Jan
Wed 7 Jan
Thu 8 Jan
Fri 9 Jan
Sat 10 Jan
Sun 11 Jan
Mon 12 Jan
Tue 13 Jan
Wed 14 Jan
Thu 15 Jan
Fri 16 Jan
Sat 17 Jan
0 6 12 18
Forumium A Web Based Forum Platform - Software Engineering - page7


23 Dec
Wed 24 Dec
Thu 25 Dec
Fri 26 Dec
Sat 27 Dec
Sun 28 Dec
Mon 29 Dec
Tue 30 Dec
Wed 31 Dec
Thu 1 Jan
Fri 2 Jan
Sat 3 Jan
Sun 4 Jan
0 6 12 18
Forumium A Web Based Forum Platform - Software Engineering - page6

Agile Model


Planning
Testing
Req Analysis
Iteration 1
Building
Designing
2-3 Months
Planning
Testing
Req Analysis
Iteration 2
Building
Designing
2-3 Months
Planning
Testing
Req Analysis
Iteration 3
Building
Designing

Software Metrics

Software metrics are quantitative measurements that are used to evaluate several facets of a software product, the project as a whole and the development process. These metrics offer useful data for assessing the performance, quality, and advancement of software development. Software metrics are crucial since they serve as a gauge for improving software processes. Software process improvement makes use of software metrics to analyze how the organization aims to accomplish the improvement objectives and how the existing process model may be enhanced. The project metrics are being used to monitor any risks, evaluate the project team's control over the quality of the software work products, and analyze the project's current state. We are using the project metrics to assess the status of our project, track potential risks and evaluate the project team’s ability to control the quality of the software work products.

Project metrics consist of several types namely size-oriented metrics, function-oriented metrics, object-oriented metrics, use-case oriented metrics and web engineering project metrics. Both size-oriented and function-oriented metrics will be used in this project. Size-oriented metrics measure project data, cost, and effort depending on the size of software created. The capability that this program will provide is the foundation for function-oriented metrics. It is an indirect metric that makes use of function points (FP), which can be calculated early in the software development process before the actual lines of code are known. FP can be obtained from a software's needs.

Each statistic has benefits and drawbacks of its own. Advantages of using size-oriented metrics are that it is easily calculated, and useful for projects with similar environments whereas the advantages of using function-oriented metrics is that it is language independent and based on data known early in the project. On the other hand, the disadvantages of size-oriented metrics are that it is language and programmer dependent whereas the disadvantages of function-oriented metrics are that it is more difficult to calculate, difficult to collect data and FP has no physical meaning.

Size Oriented Metrics

Web Application Lines of Code: 5400

Database Lines of Code: 6000

Total Lines of Code/Size: 11.4 kLOC

Project	**Effort (person-month)**	**Cost (RM)**	**Size(kLOC)**	**Doc. (pages)**	**Errors**	**People**
Forumium - Alpha 1	28	500,000	12.5	700	88	11
Forumium – Beta 2	17	360,000	11.7	650	73	8
Forumium – Gamma 3	13	210,000	11.4	620	61	5

Metrics	**Forumium - Alpha 1**	**Forumium – Beta 2**	**Forumium – Gamma 3**
Productivity = Size/Effort	12.5/28 = 0.44	11.7/17 = 0.69	11.4/13 = 0.88
Quality = Errors / Size	88/12.5 = 14.4	73/11.7 = 6.2	61/11.4 = 5.3
Documentation = Pages / Size	700/12.5 = 56	650/11.7 = 55	620/11.4 = 54

Function-Oriented Metrics

The software information domain values have been constructed by our project analysts after analysing the number of functions based on the parameters listed as the following:

Parameter	Count	Simple	Average	Complex
Inputs	7	8	10	13
Outputs	14	12	7	10
Inquiries	8	8	13	9
Files	11	9	8	12
Interfaces	6	5	9	7

Calculating the raw FP value

	Simple	**Average**	**Complex**
Inputs x Count	8 x 7 = 56	10 x 7 = 70	13 x 7 = 91
Outputs x Count	7 x 14 = 98	7 x 14 = 98	10 x 7 = 70
Inquiries x Count	8 x 8 = 64	13 x 8 = 104	9 x 8 = 72
Files x Count	9 x 11 = 99	8 x 11 = 88	12 x 11 = 132
Interfaces x Count	5 x 6 = 30	9 x 6 = 54	9 x 6 = 54
Raw FP	347	414	419

For each complexity adjustment factor, a rating on a scale of 0 to 5 is given, as shown below:

Complexity adjustment factors	**Rate**
F1: Does the system require reliable backup and recovery?	5 - Essential
F2: Are data communications required?	4 - Significant
F3: Are there distributed processing functions?	4 - Significant
F4: Is performance critical?	4 - Significant
F5: Will the system run in an existing, heavily utilized operational environment?	3 - Average
F6: Does the system require on-line data entry?	5 - Essential
F7: Does the on-line data entry require the input transaction to be built over multiple screens or operations?	3 - Average
F8: Are the master files updated on-line?	4 - Significant
F9: Are the inputs, outputs, files, or inquiries complex?	5 - Essential
F10: Is the internal processing complex?	3 - Average
F11: Is the code designed to be reusable?	5 - Essential
F12: Are conversion and installation included in the design?	3 - Average
F13: Is the system designed for multiple installations in different organizations?	4 - Sigificant
F14: Is the application designed to facilitate change and ease of use by the user?	4 - Average
SUM (CAV)	56

Calculating the FP of each Simple, Average & Complex using the formula below:

FP = Raw FP x [0.65 + (0.01 X CAV)]

- Simple: FP = 347 x [0.65 + (0.01 x 56)] = 419.87
- Average: FP = 414 x [0.65 + (0.01 x 56)] = 500.94
- Complex: FP = 419 x [0.65 + (0.01 x 56)] = 506.99

Calculating the productivity, quality, cost, and documentation of the following projects:

Project	**Effort (person-month)**	**Cost (RM)**	**Size(kLOC)**	**Doc. (pages)**	**Errors**
Forumium - Alpha 1	28	500,000	12.5	700	88
Forumium – Beta 2	17	360,000	11.7	650	73
Forumium – Gamma 3	13	210,000	11.4	620	61

Alpha 1

Metrics	**Simple**	**Average**	**Complex**
Productivity	419.87/28 = 14.99	500.94/28 = 17.89	506.99/28 = 18.10
Quality	88/419.87 = 0.21	88/500.94 = 0.18	88/506.99 = 0.17
Cost	500k/419.87 = 1190.84	500k/500.94 = 998.12	500k/506.99 = 986.21
Documetation	700/419.87 = 1.67	700/500.94 = 1.40	700/506.99 = 1.38

Beta 2

Metrics	**Simple**	**Average**	**Complex**
Productivity	419.87/17 = 24.7	500.94/17 = 29.47	506.99/17 = 29.82
Quality	73/419.87 = 0.174	73/500.94 = 0.145	73/506.99 = 0.14
Cost	360k/419.87 = 857.41	360k/500.94 = 718.65	360k/506.99 = 710.07
Documetation	650/419.87 = 1.54	650/500.94 = 1.30	650/506.99 = 1.28

Gamma 3

Metrics	**Simple**	**Average**	**Complex**
Productivity	419.87/13 = 32.30	500.94/13 = 38.53	506.99/13 = 38.99
Quality	61/419.87 = 0.145	61/500.94 = 0.12	61/506.99 = 0.12

Cost	210k/419.87 = 500.15	210k/500.94 = 419.21	210k/506.99 = 414.21
Documetation	620/419.87 = 1.476	620/500.94 = 1.23	620/506.99 = 1.22

7.0 - Project Risk Management

Step 1: Possible risk identification

The risks for our projects can be categorized into several categories namely business risks, technical & technology risks, employee risks, development environment risks and customer risks

Business Risks (BR)

- Effect on Company Revenue

  The software will be supplied exclusively to the company and will be charged per customer. This will increase revenue for the company. Profits from this project will be used to compensate the employees.

- Delivery Deadline

  The delivery deadline for this project which is 24 October is fairly reasonable given that the scope of the project remains unchanged and no changes made to the initial plan.

- End Users

  The targeted end users are the general public ranging from various age groups. Therefore, the web application should be made user friendly.

- Documentation that must be provided for customer

  Well written and informative documentation should be prepared for the customer as a manual to operate the web application developed. This is because the customer will be given full access to the database, web application, etc.

- Cost of Late Delivery

Late delivery of the application may result in customer dissatisfaction which may either lower our revenue or provide a choice for customers whether or not to accept the web application which is quite risky.

- Cost of Defective Product Delivery: Unknown

Technical & Technology Risks (TTR)

- *Are there specific methods used for software analysis?*

There are numerous methods that will be used to track and analyse the software progress regularly. Besides that, there will be frequent tests and reviews done to ensure the software is up to expectations. It is further explained in the Software Quality Assurance and Software Configuration Management document.

- *Are there any specific software tools being used?*

None confirmed yet.

- *Are there metrics to be collected in the project?*

Yes, particularly size-oriented metrics and function-oriented metrics

Is the technology something new to the organization?

No, the employees have experience working with the technology needed in this software project.

- *Do the requirements need new algorithms or I/O technology?*

None for now as in our planning, this project can be completed with the current technology available.

- *Do the requirements need new analysis, design or testing methods?*

None for now.

- *Do requirements put excessive performance constraints on the product?*

No

Customer Risks (CR)

- *Does the customer have a solid idea of what is required?*

  -Yes. In a thorough discussion, both parties mutually agreed to the same idea/concept and it was stated in the Software Requirements Specification of the project.

- *Is the customer willing to spend time in formal requirements gathering meetings to identify project scope and participate in technical reviews?*

  -Unknown. No inquiry has been made regarding to this matter.

- *Does the customer understand the software process and be familiar with the product area?*

  -Yes, the customer had attended a Software Engineering course and had deployed many web applications in the past.

Development Environment Risks

- *Are there any sophisticated software tools to be used in this project?*

-None for now.

- *Does the environment make use of the database or a repository?*

-Yes, it does.

- *Have members of the project team received training in each of the tools?*

-No software tools are identified to be used for now. However, the members are well-versed with software tools such as Github, Embold and others

- *Is there help available in using the tools if needed?*

-Yes, there are many resources available online but no software tools are being used currently in our plan.

Employee Risks (ER)

- *Are the best people available?*

-Yes, the most talented software developers, designers, and programmers are on this project

- *Do the people have the required skills*

-Yes.

- *Are there enough people available for this project?*

-Yes. A small team has been assembled for this project. This improves communication efficiency as larger teams may not be able to communicate effectively hence reducing productivity.

- *Are all the staff committed to the work?*

-Yes, they are full-time employees at Flarum with long-term contracts remaining.

- *Are the staff well-trained?*

-Yes

Step 2: Risk Table

Risk Summary	Risk Category	Probability	Impact (1–4)	RMMM (Mitigation + Monitoring + Management)
Slow page load or DB lag	Performance Risk TTR	Medium	3	*Mitigate:* Optimize queries, implement caching. *Monitor:* Track response times. *Manage:* Add server resources if needed.
Hard to maintain after deployment	Support Risk TTR	Low	2	*Mitigate:* Write documentation, follow clean coding. *Monitor:* Code reviews. *Manage:* Assign a support team member.
Features not completed on time	Schedule Risk ER	High	4	*Mitigate:* Break tasks smaller, weekly progress checks. *Monitor:* Gantt chart + daily updates. *Manage:* Extend deadline or add extra manpower if available.
Late Delivery	BR		1	
End Users Resist System	BR		1	
Changes in Requirements	CR		1	
Lack of Database Stability	TTR		3	
Poor Quality Documentation	BR		2	

Deviation from Software Engineering Standards	TTR		3	
Poor Comments in Code	ER		2	
Extra finance Required	BR		2	

Step 3: Risk Mitigation, Monitoring and Management

Late Delivery

Mitigation

Late delivery may be quite costly to the company in terms of finances and reputation. Therefore, any late delivery would be catastrophic in our point of view. Steps have to be taken for proper delivery of the software product by the deadline. One of the steps we are going to take is to estimate the project scope and set the deadline by assigning tasks to the team while taking into consideration the number of members available for the project including their skills, strengths and weaknesses. It is vital for team members to inform about their progress and address all problems that they have encountered.

Monitoring

A schedule has been established during the planning stage for project monitoring purposes. This schedule will be followed closely by the team and the potential for late delivery is indicated by falling behind schedule.

Management In the worst possible case that the project cannot be completed within the stipulated time, the only possible solution would be to request an extension from the customer which could be detrimental to the company reputation and customer satisfaction & relationship.

End Users Resist System

Mitigation

In efforts of preventing this risk, the web application will be developed with keeping end-users in the centre of our minds. The ease of use of the UI should be made our top priority.

Monitoring

Various external opinions will be gathered throughout the development process and their opinions will be further discussed by the developers in the effort of producing the best user interface possible for the end-users.

Management

In the case that the application failed to be accepted by the end-users, detailed research should becarried out to find reasons for its failure. The results could be used to make amendments to the application and release the new solution.

Changes in Requirements

Mitigation

In efforts to prevent this risk, throughout the project, there will be frequent meetings with the customer for better understanding between both parties involved. This ensures our product will meet the customer's expectations.

Monitoring

The meeting minutes of each of the meetings held with the customer should be prepared and managed properly from both sides. Then, both parties should agree to the meeting minutes before the beginning of the next meeting. Through this method, the customers and developers will bewell informed about the project.

Management

If there are any changes in requirements, immediate steps should be taken to solve this problemas quickly and effectively as possible. A meeting should be held between the project team and the customer to rectify this issue.

Lack of Database Stability

Mitigation

The developer who is mainly in charge of the database should always perform error checking. Upon determining the errors, the issues should be notified to the team and resolved at the earliest.

Monitoring

Any user who accesses the database should not perform any major modifications without the supervision of the developer who takes care of the database. At any moment, if an error is noticed, it should be informed to the team.

Management

If instability is detected with the database, the team should have a meeting to identify the potential causes and find possible solutions to the issue.

Poor Quality Documentation

Mitigation

The documentation team or manager should constantly be in contact with the developers' team. In order, to ensure that the documents are developed in a timely manner, the project manager is required to monitor the definition of document standards and mechanisms. Regular meetings should be held regarding the documentation. Any doubts should be clarified instantly. Through these meetings, if a developer finds out if any feature is missing in the documentation, it could be added too after consideration. Besides that, the testing team should do the test using this documentation and let the documentation team know about the quality of the documentation prepared for the software.

Monitoring

The development team and the testing team should always watch out for any topics about the software which may have not been included.

Management

If poor quality is detected, there should be an attempt made by the entire team to improve the quality of documentation by adding certain required topics or removing any redundant topics.

Deviation from Software Engineering Standards

Mitigation

It is very unlikely to occur since the team involved in this project is quite experienced and have a thorough understanding of the software engineering standards.

Monitoring

Technical reviews will be held regularly to ensure the documentation and actual project is consistent with each other. All the documents needed should be completed as accurately as possible which guarantees the team's compliance to software engineering standards.

Management

To handle this issue, the project must be modified in a way that it complies with the standards. Technical reviews can be used to determine how to make the project in line with set standards.

Poor Comments in Code

Mitigation

Certain commenting standards should be established by the lead programmer to ensure good quality code is produced.

Monitoring

The code should be review and the reviewers will determine if the comments given for the code is appropriate. This ensures that good quality code is created.

Management

If the code quality is detected to be low, then the programmers should perform the necessary actions to improve the code. Constant close monitoring of the code will reduce the possibility of poor comments in the code. Programmers could resolve this issue by simply modifying the comments placed or adding new comments.

Extra Finance Required

Mitigation

In future, due to unforeseen circumstances or improper planning, there may be a need for extra money. As a precautionary measure, there could be a reserve fund allocated for such unexpected scenarios.

Monitoring

The finance team should always monitor the cash flow associated with the project and alert the project team in the event of exceeding the allocated budget.

Management

In the case that extra finance is required, the project manager should have a meeting with the project sponsor and request extra funds. However, this is very unlikely to happen given that the team have handled few similar projects successfully.

Step 4: Risk Map

Risk Map					
Level of Likelihood	Almost Certain	🟧	🟧	🟫	🟥
	Likely	🟦	🟧 TTR3	🟫 BR3 TTR1 ER2	🟫
	Possible	🟩	🟦	🟧	🟫 BR1
	Unlikely	🟩 ER1	🟩	🟦 BR4, CR1	🟧 BR2
	Rare	🟩	🟩 TTR2, TTR4	🟦	🟧
		4- Negligible	3- Marginal	2- Critical	1- Catastrophic
		Level of impact			

Legend	🟩	🟦	🟧	🟫	🟥
Risk Rating	Low	Moderate	Significant	High	Extreme

Business Risk (BR)			
BR1	Late Delivery	High	P-Possible, 1-Catastrophic
BR2	End User Resist System	Significant	U-Unlikely, 1-Catastrophic
BR3	Poor Quality Documentation	High	L-Likely, 2-Critical
BR4	Extra Finance Required	Moderate	U-Unlikely, 2-Critical
Customer Risk			
CR1	Changes in Requirement	Significant	U-Unlikely, 2-Critical
Technical and Technology Risk (TTR)			
TTR1	Lack of Database Stability	High	Likely, 2-Critical
TTR2	Deviation from Software Engineering Standards	Low	Rare, 3-Marginal
TTR3	Hard to maintain after deployment	Moderate	Likely, 3-Marginal
TTR4	Slow page load or DB lag	Low	Rare, 3-Marginal
Employee Risk (ER)			
ER1	Poor Comments in Code	Low	U-Unlikely, 4-Negligible
ER2	Features not completed on time	Significant	Likely, 2-Critical

Risk Profile

Risk	Likelihood	Impact	Risk Rating	Mitigation Plan
Business Risk				
BR1 - Late Delivery Scope: Late delivery may be quite costly to the company in terms of finances and reputation.	P-Possible Justification: An inadequate number of team members. Large project job scope.	1- Catastrophic Justification: Effects on the reputation. Service shortfall incurs a penalty.	High	Estimate the project scope and set the deadline by assigning tasks to the team while taking into consideration the number of members available for the project including their skills, strength and weaknesses.
BR2 - End User Resist System Scope: The final product is undesired by the customer.	U-Unlikely Justification: The client signed the agreement and accepted the tender	1- Catastrophic Justification: This will greatly postpone the application work schedule.	Significant	Development should keep end-users in the centre of our minds. The ease of use of the UI should be made our top priority.
BR3 - Poor Quality Documentation Scope: Documentation work, if any, are unrelated to the overall project.	L-Likely Justification: The company prioritises working product rather than proper documentation. Early stages can be ad-hoc.	2-Critical Justification: Poor documentation will lead to trouble in audit and testing stages.	High	The documentation team will work closely with the developers' team. Regular meetings held to monitor the definition of documentstandards and mechanisms.
BR4 - Extra Finance Required Scope: Change requirements, late deadlines will increase cost of project.	U-Unlikely Justification: The cost budget is planned out according to requirements. The finance team monitors the cash flow.	2-Critical Justification: Without cash flow, the project schedule will be delayed.	Moderate	As a precautionary measure, there could be a reserve fund allocated for such unexpected scenarios.
Customer Risk				

CR1 - Changes in Requirements Scope: Clients may request for huge changes that overhauls the project.	U-Unlikely Justification: Tender project is clearly defined by the client side.	1-Catastrophic Justification: Sudden changes of requirement increases cost and delays final product launch.	Significant	Frequent meetings held with the customer for clear communication.
Technical and Technology Risks				
TTR1 - Lack of Database Stability Scope: Database has to handle occasional user spikes, while not sacrificing its efficiency and speed.	L-Likely Justification: Database deployment requires a backend team, with clear communication with the rest of the developers.	2-Critical Justification: A web app without the database is as good as a sitting duck.	High	Perform error checking regularly. Upon determining the errors, the issues should be notified to the team and resolved at the earliest.
TTR2 - Deviation from Software Engineering Standards Scope: Software Engineering is a discipline that practices the best way to develop software.	R-Rare Justification: The management and developer teams have been well-trained in regards to standards compliances.	3-Marginal Justification: The project's main concerns are proper documentation and a highly reliable working product.	Low	Have a thorough understanding of the software engineering standards.
TTR3 - Hard to maintain after deployment Scope: It specifically affects the engineering team's ability to fix bugs, implement new features, and scale the platform as the user base grows.	Likely Justification: Forums often rely on third-party plugins or rapid prototyping to get features live, leading to "spaghetti code." Its point to Rapid Development Cycles	3 - Marginal Justification: While maintenance difficulties are frustrating for developers, the immediate impact on the end-user is low.	Moderate	Shift toward a microservices or modular monolith approach. This ensures that the "User Authentication" module is independent of the "Forum Categorization" module.

TTR4 - Slow page load or DB lag Scope: Latency experienced by the end-user. It covers front-end rendering JavaScript/CSS weight and back-end bottlenecks, such as unoptimized SQL queries, lack of indexing, or server resource exhaustion during peak traffic.	Rare Justification: For a new platform, the volume of data and concurrent users is typically well within the hardware's capabilities.	3 - Marginal While a slow site is annoying, it is considered "marginal" because it does not result in data loss or security breaches. Users might experience a "jerkier" interface or wait an extra second for a thread to open, but the core utility of the forum remains intact.	Low	To ensure performance remains high even as the community grows by implementing lazy loading for long forum threads so that only the first few comments load initially, reducing the initial payload.
Employee Risks				
ER1 - Poor Comments in Code Scope: The comments in the code poorly explained the functionality enough.	U - Unlikely Justification: The developers practice a standard of commenting.	4-Negligible Justification: The source code is not the priority of the client.	Low	Lead programmer establishes a commenting standard that will be followed by the rest of the developers.
ER2 - Features not completed on time Scope: Involves the product roadmap and the development lifecycle. This includes the design, coding, testing, and deployment phases for specific milestones	L - Likely Justification: Software complexity is often underestimated; unforeseen technical hurdles like integrating a specific API or fixing a complex bug which frequently extend timelines.	2 - Critical	Significant	Break down large tasks into smaller, manageable two-week sprints. This allows for frequent progress checks and early detection of delays using Agile or Sprig Methodology

Conclusion

Project Risk Rating: High Risk

Contributing Factors

a) Inadequate number of team members combined with a large project job scope.

b) Company culture prioritizes a working product over proper documentation; early stages tend to be ad-hoc.

c) Database deployment requires a dedicated backend team, necessitating clear communication with all other developers.

Recommendations

a) Estimate the total project scope and establish a realistic deadline by assigning specific tasks to the team while accounting for the current headcount.

b) Integrate the documentation team closely with the developers; hold regular meetings to bridge the gap between code and records.

c) Perform database error checking regularly and resolve any identified issues at the earliest possible stage.

d) Set up a reserved fund as a precautionary financial measure.

Key Areas for Attention

a) Legal Guardrails: Contract T&Cs must include a review of penalty clauses, inclusion of limitation of liability, clear conditions/methodology for penalty imposition, and defined reciprocal responsibilities.

b) Stakeholder Clarity: Define clear roles and responsibilities between the Flarum platform team and the client.

c) Commercial Health: Ensure the profitability and long-term sustainability of the chosen business model.

d) Governance: Maintain effective project management throughout the lifecycle.

e) Oversight: Implement rigorous project progress tracking, frequent reviews, and active monitoring.

Software Cost Estimation and Risk Analysis

Task List includes:

- Set up project environment and development
- Design database and source code
- Implement core features (posting, replying, user accounts)
- Test everything and rework
- Final Deployment
- Time Estimation

Activity	**Immediate Predecessor**	**Optimistic (O)**	**Most Likely (M)**	**Pessimistic (P)**	**Duration E (days)**
Set up environment	–	3	6	10	**6**
Design database	1	6	10	16	**10**
Implement core features	2	13	23	32	**23**
Test features	3	3	6	13	**7**
Deploy project	4	2	3	7	**4**
Total					**50 days**

Impact Level:

1 - Catastrophic 2 - Critical 3 - Marginal 4 - Negligible

For the time estimation for the Risk Analysis focuses on the Development, Testing, and Deployment phases, as these stages carry the highest uncertainty and potential impact to the project schedule.

flarem plus.

SOFTWARE QUALITY ASSURANCE (SQA) DOCUMENT

1. Introduction

1.1 Purpose

This Software Quality Assurance (SQA) Plan defines the quality processes, standards, and controls to be applied during the development and deployment of **FlaremPlus**, a web-based online forum platform for university communities. The purpose of this plan is to ensure that FlaremPlus meets functional requirements, usability expectations, performance needs, and reliability standards while maintaining ease of access and long-term maintainability.

1.2 Scope

This SQA Plan applies to all phases of the Software Development Life Cycle (SDLC) of FlaremPlus, including requirement analysis, system configuration, customization of the forum platform, testing, deployment, and maintenance. The plan covers both functional and non-functional quality aspects such as usability, accessibility, security, and performance.

1.3 Problem Overview

Currently, universities rely on multiple communication tools such as WhatsApp, Instagram, and email newsletters. These tools are not designed for structured discussion or information retrieval, resulting in missed messages, repeated questions, and low engagement. Existing university portals focus mainly on administrative functions and lack community-driven interaction. FlaremPlus addresses this gap by providing a centralized, searchable, and topic-based discussion platform for students and instructors.

1.4 Definitions and Acronyms

- **SQA**: Software Quality Assurance
- **SDLC**: Software Development Life Cycle
- **UAT**: User Acceptance Testing
- **CMS**: Content Management System

2. SQA Management

2.1 SQA Organization

The SQA activities for FlaremPlus are managed by the project team, consisting of a Project Manager, Software Quality Assurance Manager, developers, and testers. Quality assurance tasks are integrated throughout the development and deployment stages of the system.

2.2 Roles and Responsibilities

- **Faris Amsyari Bin Saiful Fazamil-Project Manager:** Oversees project execution and ensures quality objectives are achieved within schedule constraints.
- **Akmal Danish Jasni-Lead Programmer :** Oversees system development tasks, reviews code, and ensures technical requirements are met.
- **Mohamad Syafiq Haikal-Software Quality Assurance Manager:** Defines quality standards, monitors compliance, and conducts reviews and audits.
- **Mir Mohammad Muaz-Developer:** Customize and configure the Flarum platform, follow coding and configuration standards, and perform unit testing.
- **Muhammad Zaidani Bin Mohd Ezri-Sub Programer:** Works together with the lead programmer

2.3 SQA Tasks

- Review system requirements and configuration settings
- Ensure adherence to web development and security standards
- Conduct periodic reviews and quality audits
- Ensure that software quality (SQ) personnel have adequate workspace and the necessary tools to perform SQ activities.
- Offer overall guidance and direction to SQ staff responsible for executing software quality tasks and assessments.
- Deliver weekly and quarterly software quality status updates.

- Support SQ personnel in addressing and resolving any noncompliance issues, problems, or risks identified during SQ activities.
- Report any unresolved noncompliance matters to project management for escalation.

3. Documentation Standards

All project documents shall follow approved templates and standards.
Documentation includes: - Software Requirements Specification (SRS) Software Design Document (SDD) - Test Plan and Test Cases - User Manuals

Documents shall be reviewed and approved before proceeding to the next SDLC phase.

4. Standards, Practices, and Conventions

4.1 Development and Configuration Standards

FlaremPlus is built using the open-source forum platform **Flarum**, which is based on PHP and follows a modular architecture. All customizations and extensions shall comply with Flarum development guidelines and PHP best practices.

4.2 Web and Design Standards

The system shall follow responsive web design principles to ensure accessibility across desktops, tablets, and mobile devices. User interfaces shall be simple, lightweight, and easy to navigate to encourage participation.

4.3 Testing Standards

Testing activities shall follow defined procedures covering functional testing, usability testing, performance testing, and security testing.

5. Software Reviews and Audits

Formal reviews shall be conducted at each major SDLC phase: Requirements Review - Design Review - Code Review - Test Review

Audits are conducted to verify compliance with approved processes and standards.

6. Software Testing

6.1 Test Strategy

The testing strategy focuses on ensuring that FlaremPlus supports structured discussions, easy information retrieval, and stable performance under multiple users. Testing is conducted throughout development to identify defects early.

6.2 Types of Testing

□ Unit Testing: Verification of individual components and extensions

- **Integration Testing**: Validation of interactions between Flarum, MariaDB, and NGinx
- **System Testing**: End-to-end testing of forum features such as posting, commenting, and searching
- **User Acceptance Testing (UAT)**: Validation by representative students and instructors

6.3 **Defect Management**

All identified defects shall be logged, prioritized, and tracked until resolution. Corrective actions shall be verified before closure.

7. Configuration Management

Configuration management ensures consistency and control over all FlaremPlus software components. Version control tools are used to manage configuration files, extensions, and documentation. The deployment environment includes: - **Web Server**: NGinx - **Backend Platform**: PHP (Flarum) - **Database**: MariaDB

All changes shall be documented and approved before deployment.

8. Risk Management

Potential project risks related to quality shall be identified and mitigated.
Regular risk assessments shall be conducted throughout the project lifecycle.

9. Training

Project members shall receive necessary training on tools, standards, and quality procedures to ensure effective implementation of the SQA plan.

10. Metrics and Measurement

Quality metrics such as defect density, test coverage, and review effectiveness shall be collected and analyzed to monitor software quality.

11. Problem Reporting and Corrective Action

Non-conformances shall be documented, analyzed, and corrective actions shall be implemented to prevent recurrence.

12. Tools and Techniques

The following tools may be used: - Version control systems (e.g., Git) -
Testing tools - Issue and defect tracking tools

13. Approval

This SQA Plan shall be reviewed and approved by project stakeholders before implementation.

Prepared by: Software Quality Assurance Team

Date: ____30/12/2025________________

Signature Page

Prepared by,

Mohamad Syafiq Haikal Bin Abdul Aziz

30th December 2025

Software Quality Assurance Manager

Reviewed by,

Akmal Danish Jasni

10th January 2026

Lead Programmer

Reviewed by,

MuazMir

Mir Mohammad Muaz

10th January 2026

Developer

Approved by,

Faris Amsyari Bin Saiful Fazami

10th January 2026

Project Manager

flarem plus.

SOFTWARE CONFIGURATION MANAGEMENT (SCM) DOCUMENT

1. Identification

1.1 Document Overview

This report is a Software Configuration Management plan of Forumium, which is a web based forum administration that was created by Flarem Plus Software. This document is aimed at defining a formal and methodical process of locating, managing, monitoring, and auditing all software configuration objects within the software development life cycle.

Software Configuration Management is an assurance that the Forumium system is consistent, traceable, and stable as it is developed, undergoes testing, deployment, and maintenance. The SCM operations outlined in this document involve configuration identification, configuration control, version management, configuration status accounting, and configuration auditing.

1.2 Abbreviations and Glossary

1.2.1 Abbreviations

Abbreviation Definition;-

SCM – Software Configuration Management

SCI – Software Configuration Item

ECP – Engineering Change Proposal

CCA – Configuration Control Authority

PDR – Preliminary Design Review

CDR – Critical Design Review

TRR – Test Readiness Review

1.2.2 Glossary

Baseline - A formally reviewed and approved version of a configuration item or group of items that serves as a reference for further development and can only be modified through formal change control.

SCI - A software related artifact such as source code, configuration files, or documentation that is managed under configuration control.

Identification - The process of uniquely naming and organizing configuration items within the SCM repository.

Change Control - A controlled procedure that ensures modifications to configuration items are evaluated, approved, and documented before implementation.

Version - A uniquely identified state of a configuration item that reflects functional or structural changes.

Revision - A modification applied to a version to correct defects without altering documented functionality.

Release - The formal distribution of an approved software version.

Configuration Audit - A systematic review to verify that configuration items comply with approved baselines and requirements.

2. Organization

The members of the Flarem Plus Software development team undertake Software Configuration Management of the Forumium project. There are responsibilities that are well delegated to achieve proper coordination, accountability and control of the project life cycle. Main positions undertaken by the SCM activities are Project Manager, Software configuration Manager, Software Quality assurance Manager, Software engineers, Software designers and client stakeholders.

A centralized SCM repository is used to manage configuration items, version history, and approved baselines. The system is also executed and tested with external development and run time environments. The official configuration items are only seen as those that are stored in the SCM repository.

2.1. Decisions Process and Responsibilities

Role	Activities and Responsibilities	Decision Responsibilities
Project Manager	Coordinates project activities and approve baselines and releases.	Final authority on change approvals and conflict resolution.
Software Configuration Manager	Manages GitHub repository, version control, and configuration items.	Evaluates change impact and configuration enforces configuration control decisions.

Software Quality Assurance Manager	Conducts audits and ensures compliance and quality.	SCM Approves quality compliance or requests rework.
Software Designer	Prepares and maintains system design documents.	Confirms design impact of proposed changes.
Programmer	Implements approved changes and supports testing.	Advises on technical feasibility of changes.
Documentation Manager	Maintains and versions project documents.	Confirms documentation readiness for approval.

3. Configuration Identification

Configuration identification forms the foundation of the SCM process for the Forumium project. All software configuration items are uniquely identified, named, and organized using an object-oriented approach within the GitHub repository.

Configuration items include source code modules, configuration files, documentation, database schemas, and deployment scripts. Each item is traceable through its identification scheme, version history, and associated change records.

3.1 Identification Rules

3.1.1 Identification Rules of Configuration Items

3.1.1.1 Identification of a Configuration Item

Each configuration item is identified using a standardized naming convention to ensure uniqueness and traceability. The general identification format used is:

“XXX_Vm.n”

Where XXX represents the configuration item name and Vm.n represents the version number.

3.1.1.2 Version Number of a Configuration Item

A version number is mandatory before any delivery or release. The Forumium project adopts a structured versioning scheme:

- Major

- Minor
- Revision

Major versions indicate significant functional changes, minor versions indicate incremental improvements, and revision numbers represent bug fixes or maintenance updates.

Official versions are marked using GitHub tags after approval and testing.

3.1.2 Identification Rules of Documents

3.1.2.1 Description of Documents Identifiers

Documents are identified using the following format:

XXX_<document type>*<document number>*<revision index>

This ensures consistent identification and controlled document evolution.

3.1.2.2 Definition and Evolution of the Revision Index

The revision index is updated prior to document distribution. Major revisions reflect substantial changes, minor revisions reflect incremental updates, and revision numbers reflect corrections or refinements.

3.1.3 Identification Rules of a Media

3.1.3.1 Internal Identification

Media used for delivery or backup purposes are identified using the following format:

<configuration item identification>/<media>/<volume>

This allows traceability when multiple media units are involved.

3.2 Reference Configuration Identification

Reference configurations are defined at key project milestones. Each reference configuration is identified by its contents, approval status, and associated review activities. These reference configurations provide stable checkpoints during development and testing.

3.3 Configuration Baseline Management

The projects of the Forumium rely on configuration baselines as a way of establishing reference points that are stable and approved throughout the software development lifecycle. After having a baseline in place, then any modifications to its contents are also regulated using the formal configuration control process.

Three baselines are defined. The functional baseline is the one that records the approved system requirements and anticipated functionalities. The design baseline comprises of accepted system designs and architecture artifacts. Product baseline includes completed source code, configuration files and documentation that is all set to be deployed.

GitHub is used as all baselines are version controlled. The Project Manager allows baseline approval, whereas the Software Configuration Manager manages baseline creation and maintenance. This will provide consistency, traceability and controlled evolution of the system across the project.

4. Configuration Control

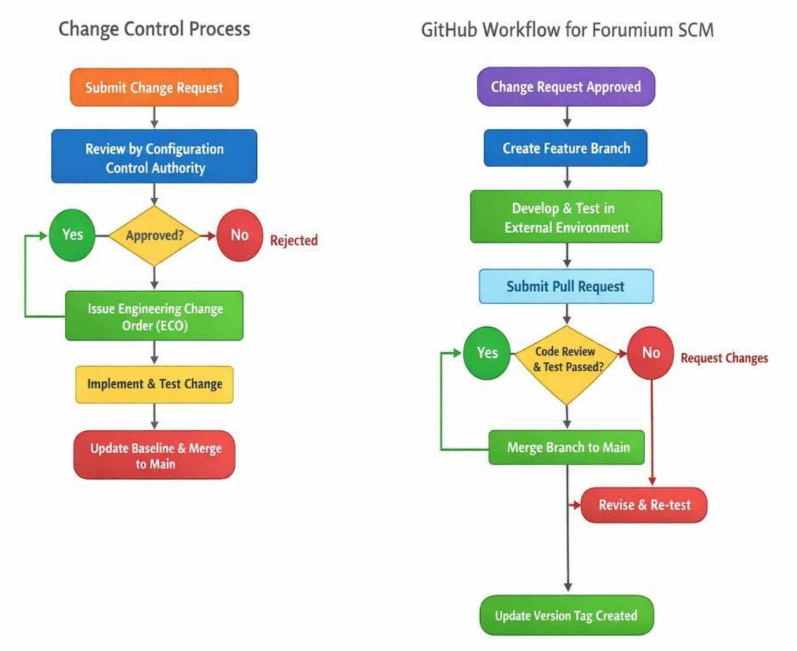
Configuration control is used to make sure that all modifications in the system of the Forumium are brought in a methodical and controlled way. Engineering Change Proposals are the documents which initiate changes, their cause, configuration items which are impacted and the effect on costs and schedule. Configuration Control Authority considers every proposal and decides to approve or reject the change.

Configuration control is applied through github branching and pull request mechanisms. Modifications on the primary branch are prohibited.

4.1 Change Control Process

The Project Manager or stakeholders initiate change requests. Once approved, a dedicated GitHub branch is created to implement the change. After development and testing in external execution environments, the change is reviewed and merged into the main branch through a pull request.

In case of a rejected change, the branch is closed and the integration is not done. Approved and merged changes are the only modifications that are considered as a part of the official configuration.


Change Control Process
Submit Change Request
Review by Configuration Control Authority
Yes
Approved?
No
Rejected
Issue Engineering Change Order (ECO)
Implement & Test Change
Update Baseline & Merge to Main
GitHub Workflow for Forumium SCM
Change Request Approved
Create Feature Branch
Develop & Test in External Environment
Submit Pull Request
Yes
Code Review & Test Passed?
No
Request Changes
Merge Branch to Main
Revise & Re-test
Update Version Tag Created

5. Configuration Support Activities

5.1 Configuration Status Accounting

Configuration Status Accounting documents and reports configuration item status during the software lifecycle. The main configuration status records of the Forumium project are github commit histories, github pull requests and tagged releases.

Execution environments are treated as non authoritative. Only artifacts committed to GitHub are recognized as official configuration items.

5.1.1 Evolutions Traceability

Every change is tracked with the help of GitHub commit identifiers. Commit messages explain the nature of changes, modules that are being changed and approvals. Document changes are tracked through revision histories.

5.1.2 Setting up Configuration Status

Each configuration item is labeled with its version number, creation date, and associated Version Delivery Description prepared by the Software Configuration Manager.

5.1.3 Configuration Status Diffusion

Configuration status information is distributed by the Software Configuration Manager and Software Quality Assurance Manager to relevant stakeholders as required.

5.1.4 Configuration Status Records Storage

All configuration records are stored within the GitHub repository, including change requests, documentation, version delivery descriptions, and historical configuration states. GitHub serves as the single source of truth for configuration records.

5.2 Configuration Audits

Configuration audits are conducted to verify that the implemented system conforms to approved baselines and requirements. Baseline audits and functional configuration audits are performed to ensure traceability, consistency, and compliance.

5.3 Reviews

Formal reviews conducted during the project include Preliminary Design Review, Critical Design Review, and Test Readiness Review. These reviews assess design completeness, implementation readiness, and testing compliance.

5.4 Configuration Management Plan Maintenance

The Software Configuration Manager maintains this software configuration Management plan. Any updates of this plan use the same configuration control and version management processes detailed in this document. Accepted changes are logged into the GitHub repository so as to maintain coherence between paperwork and system execution.

flarem plus.

SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

1.0 Introduction

1.1 Purpose and Scope

The purpose of this project is to develop a web-based forum application that enables students to communicate, share information, and engage in discussions efficiently through an online platform. The application allows admin and moderators to manage forum content and user activities to ensure a structured and positive communication environment. This system aims to enhance collaboration among students while providing a centralized space for academic and non-academic discussions.

The application is supported by a relational database that stores information related to admin, moderators, and student users. The database also manages forum post and user interactions. With a focus on accessibility and usability, students can access the forum using various devices such as computers and tablets while experiencing consistent functionality and user experience across platforms.

The forum application provides effective reporting and monitoring features, allowing admin and moderators to view user activity and manage reported content. Students can easily view the status of their posts and interactions through their user dashboard. This ensures transparency and encourages active participation within the forum.

Overall, the application allows educational institutions and student communities to efficiently manage online discussions with minimal effort. Admin can moderate, edit, or remove content with just a few clicks, while students can communicate and collaborate without relying on traditional communication methods such as emails or messaging applications. This reduces administrative workload and fosters an engaging, well-organized online communication platform for students.

Definitions, Acronyms and Abbreviations

Acronyms/Abbreviations/Words	Definition
SRS	Software Requirement Specifications
DB	Database
SCM	Software Configuration Management

1.2 Overview

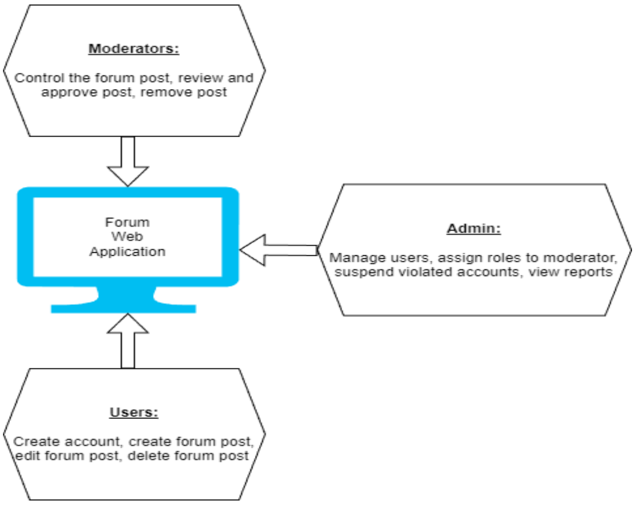
The SRS describes the overall characteristics and scope of the web-based forum application developed to support student communication and interaction. This document presents a general description of the system from both product and user perspectives, including identified constraints, assumptions, and dependencies that may affect the system's design and implementation. In addition, it details all relevant system interfaces, including user interfaces, software interfaces, communication interfaces, and any applicable hardware considerations required for the operation of the web application. The functional requirements of the system are illustrated using Use-Case Diagrams to clearly define interactions between users and the system, while the non-functional requirements such as performance, usability, security, and reliability are also specified. Analysis models are applied to further clarify and structure the system behaviour. These models serve multiple purposes, including validating requirements and ensuring that the system meets user needs once development is complete. Various diagrams are used within the analysis model to represent system processes, data flow, and interactions. The change management in the software requirements will also be explained in the later part of this document.

1.3 General Description

This document explains the challenges and issues encountered during the development of the web-based forum application, which may impact the project's progress and scalability. It describes the requirements and ideas requested by the clients and proposed solutions designed by the development team. The document also summarizes the key software features and proposed system components. The system requirements including functionality, security, supportability, constraints, and interfaces are briefly explained.

2.0 Product Perspective

The web-based forum application is specifically designed for a university to facilitate communication among students, lecturers, and administrators. This application provides functionalities like online discussion forums and academic collaboration platforms. It interfaces with a MySQL relational database connected to a web server and supporting services such as a storage server. The primary users of this application include admins, moderators, and students. To operate effectively, the system requires user management, forum and discussion management, and administrative control of the application. This system architecture is illustrated in the following figure.


Moderators:
Control the forum post, review and approve post, remove post
Forum Web Application
Admin:
Manage users, assign roles to moderator, suspend violated accounts, view reports
Users:
Create account, create forum post, edit forum post, delete forum post

Each group works with a dedicated server or module. All of them run on a local server (XAMPP or PHPmyadmin) of the web forum application. The admin and moderator module can manage and control the forum environment from each user. Every person needs their credentials to login into the system such as username and passwords.

All of them can be reached using web interfaces such as Google Chrome, Firefox, etc.

3.0 Product Functions

This product will have functionality like common online community or membership-based platforms. For example, the system is designed for use within a university environment to support communication and discussion among its members. Users of the forum will have individual accounts that allow them to create posts, participate in discussions, manage their profiles, and edit or remove their own content.

Moderators are responsible for managing discussion activities by reviewing, approving, editing, or removing posts when necessary. They can also monitor forum activity and use discussion statistics to ensure productive and appropriate communication within the platform.

The system also maintains records of university users, including students and staff, to support effective content management and user interaction. Upon registration, users are added to the system and granted access based on their assigned roles. Admins have full control over the system, including managing user accounts, assigning moderator roles, and removing or blocking users when required. Admins can also manage overall system settings to ensure smooth and organized operation of the web-based forum application.

4.0 User Characteristics

The users of this software are mainly classified in three categories:

User

- The student who are trying to find any knowledge or information in the internet via forum post
- The student may or may not be very familiar with using the web application

Moderator

- The user that will be handling mostly the web application
- Needs to have basic knowledge of using the web application and capable of handling student's forum post through the application
- The user must have common sense and not be biased toward any forum posts made and shall be able to maintain safe environments in the web application

Admin

- The user that will mainly in charge of managing the application
- Adding and deleting the users and moderators of the application
- Recruit new moderators to help manage the application.
- Aid the moderators in case of any issues.

Each user has different expectations from the web application that is being developed

User

- Wants to view, make and delete forum post
- Wants to search for specific forum post
- Wants to report on inappropriate forum post or comment

Moderator

- Wants to review and remove the inappropriate forum post
- Wants to assist the user about the web application

Admin

- Wants to have ability to be the main controller of the application
- Wants to control the access and usage capabilities of the other types of users
- Wants to assist the moderator about the application
-

5.0 General Constraints

Fully functional Internet network connection and 24-hour server uptime for optimal functionality are required for the proposed solution. In the case when there is a loss of Internet connection or server downtime during any active session from the user, an error message will be displayed such as "*temporarily lost network connection*" or "*Server not found*". The error message will depend on the occurrence of the error. Hence, to counter this issue, the user needs to perform the task again upon resumption of Internet network connection and server.

6.0 Assumptions and Dependencies

During the software development, it is assumed that the university will utilize computer hardware running a Windows operating system, specifically Windows 10 or later. It is also assumed that the university will provide the necessary server infrastructure to host the application and its database.

The system depends on the availability of at least two servers for data storage, where one server is used for primary data management and the other for data backup. A registered domain is required to allow users to access the forum through a web interface. The web application must have authorized access to the database to retrieve and store user, post, and discussion data. In addition, users accessing the forum are required to have a compatible device and a stable Internet connection to ensure proper system functionality.

7.0 Functionality and Features

7.1 Requirements

Feature 1 <Register new user / Login>

New users should be able to register a new account by entering their university's email address, password, username, gender and contact number in the register page. Then, their entity will be created in the database. For existing users, they are required to login using the credentials before they can enter forum website.

Feature 2 <Make / View / Edit / Delete a forum post>

The existing users must be able to view all the forum posts made from other users. Not only that, but they must also be able to make a new forum post. The forum made by them should be viewed, edited and deleted by themselves.

Feature 3 <Moderation>

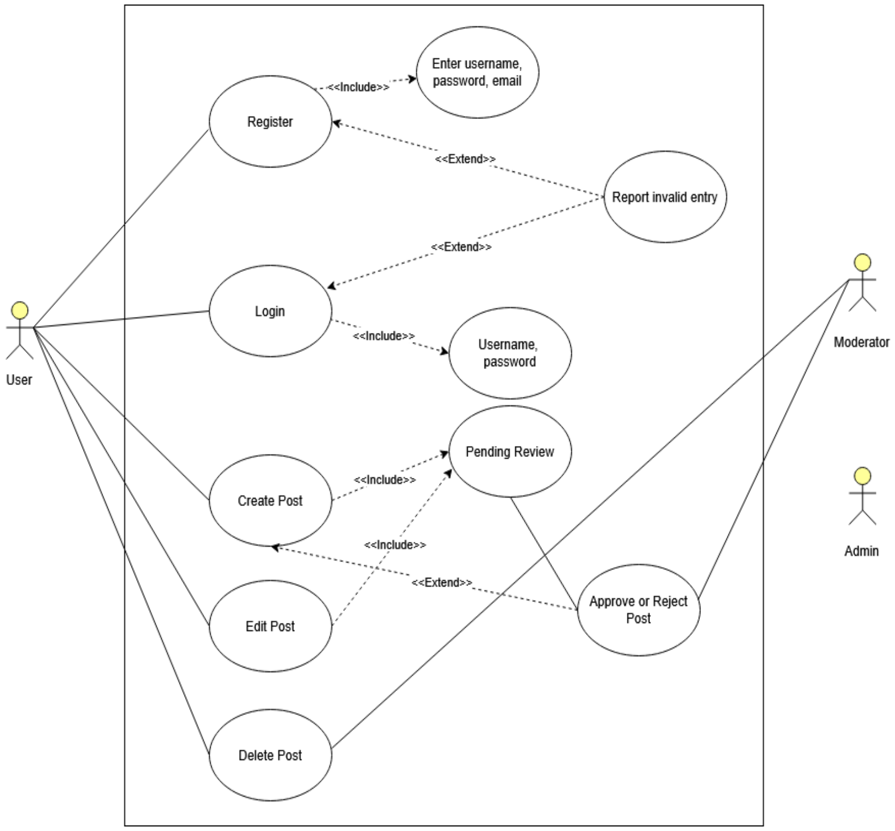
The existing users should be able to report any forum posts if they find it inappropriate or break the rules and terms and services. The reported posts will be reviewed by the moderators and will be removed if it is deemed necessary.

Feature 4 <Admin management>

The admin should be able to add, remove, edit and delete any existing users or moderators from their access. The recruitment of moderators is also made by the admin to aid them in any issues.

7.2 Use Cases

Web-based forum application

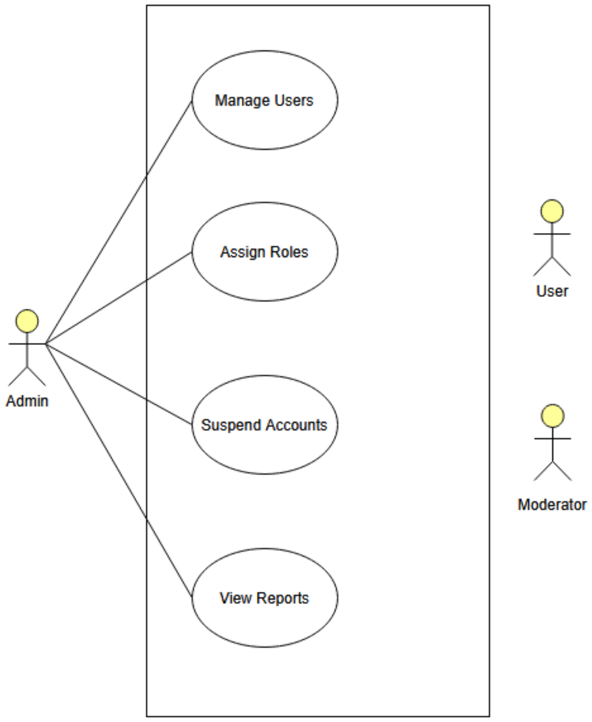


Use case diagram from users' point of view

Name/Description	User and Moderator Use Cases

Actors	1. User 2. Moderator 3. Admin
Main Success Scenario	1. The user accesses the Forum Web Platform and registers an account by entering a valid username, password, and email. After successful registration, the user logs into the system using valid credentials and gains access to the platform features. 2. The user then creates a post, which is automatically submitted for pending review. A moderator reviews the post and approves it, after which the system publishes the post and makes it visible to other users. The user may subsequently edit or delete their post as needed while using the platform.
Alternate Flows	1. If the user enters invalid or incomplete information during registration or login, the system reports an invalid entry and prompts the user to re-enter the required details. If a submitted post is rejected by the moderator, the system updates the post status to rejected and notifies the user, who may choose to edit and resubmit the post for another review. 2. At any point after creating a post, the user may also choose to delete the post, in which case the system removes it and terminates the posting process.

Web-based forum application

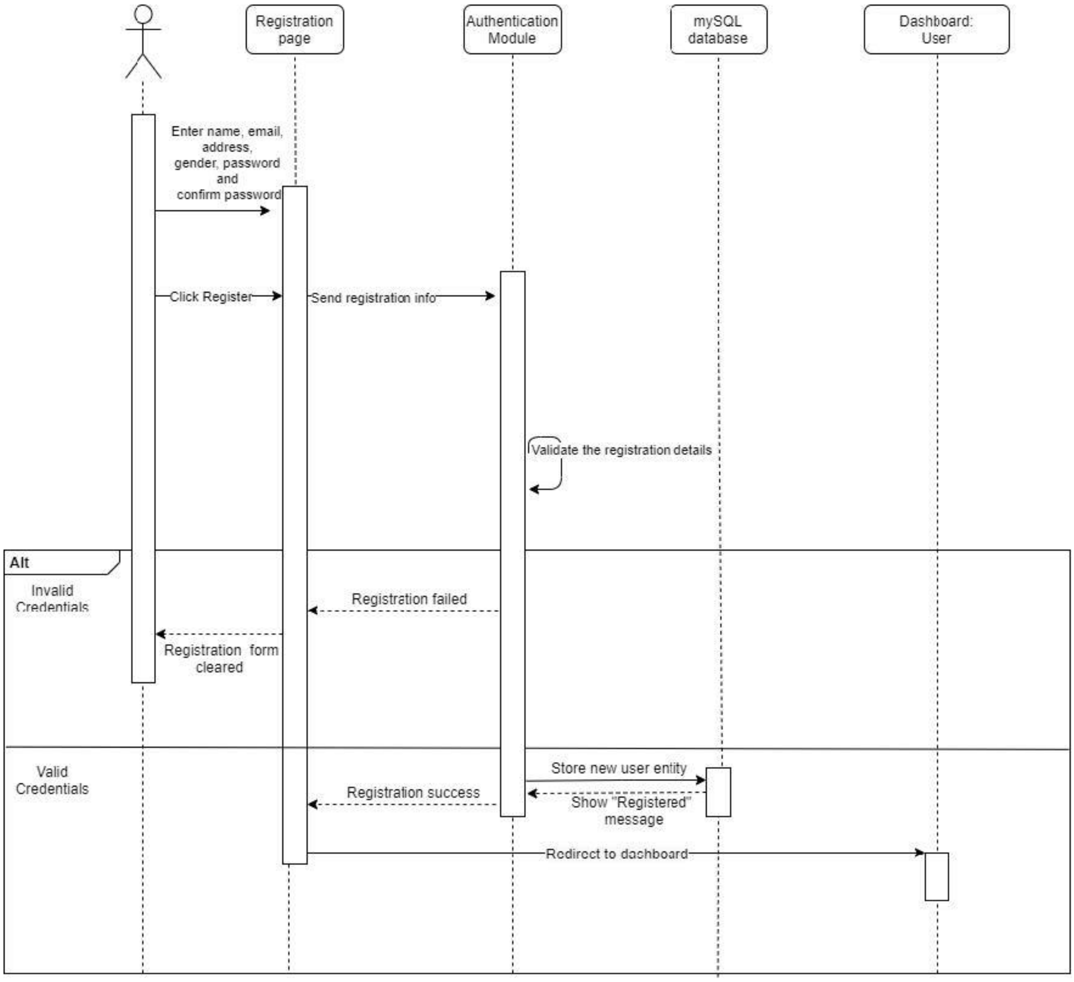


Use case diagram for admin's point of view

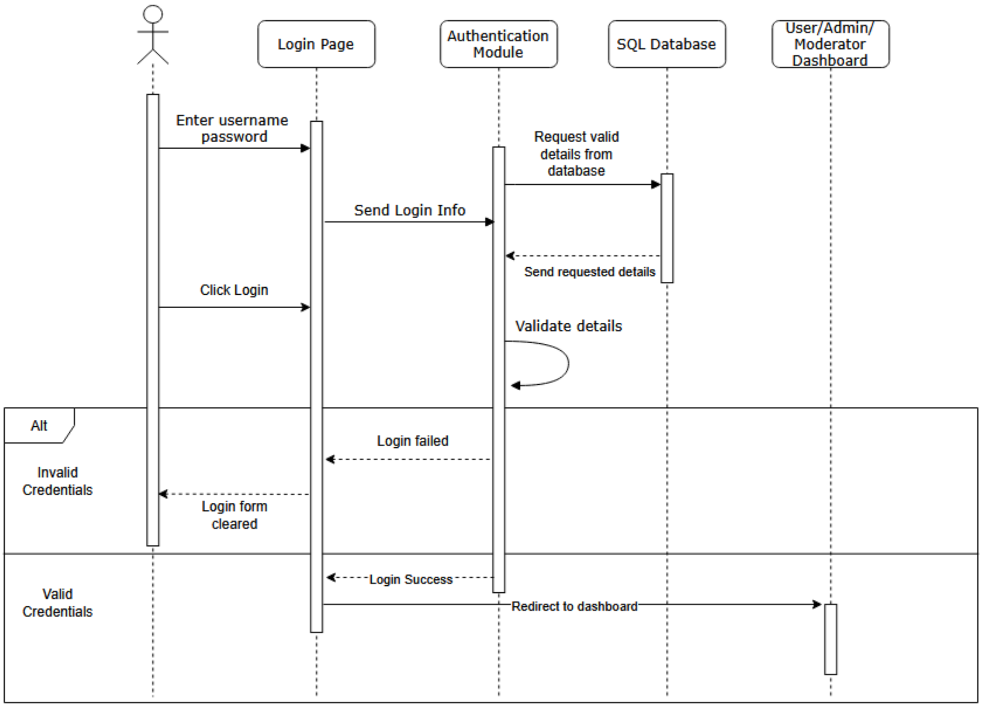
Name/Description	Admin Use Case
Actors	1. Admin 2. Manager 3. User
Main Success Scenario	1. The admin account must exist in the system and have valid administrative privileges. The Forum Web Platform and database must be operational, and the admin must be successfully authenticated before performing any administrative actions
Alternate Flows	1. If the admin attempts to manage users or assign roles to a non-existent or invalid account, the system displays an error and prevents the operation. 3. If a role assignment or suspension action cannot be completed due to system errors or insufficient privileges, the system notifies the admin and no changes are applied.

	4. When viewing reports, if report data is unavailable or a system failure occurs, the system informs the admin and allows the admin to retry or exit the function.
Pre-Conditions	1. The admin account must exist in the system and have valid administrative privileges. The Forum Web Platform and database must be operational, and the admin must be successfully authenticated before performing any administrative actions.

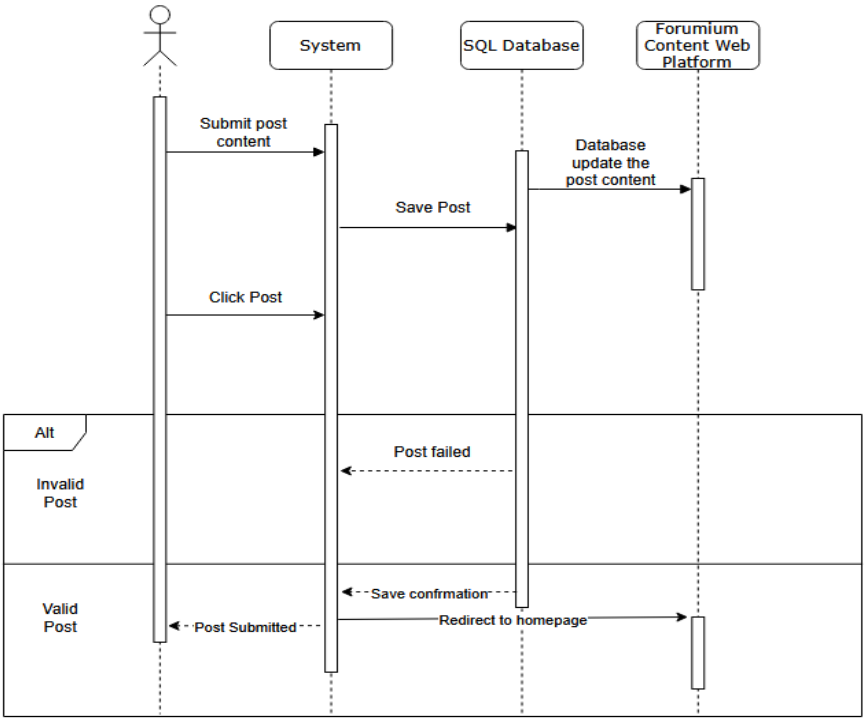
7.3 Sequence Diagrams



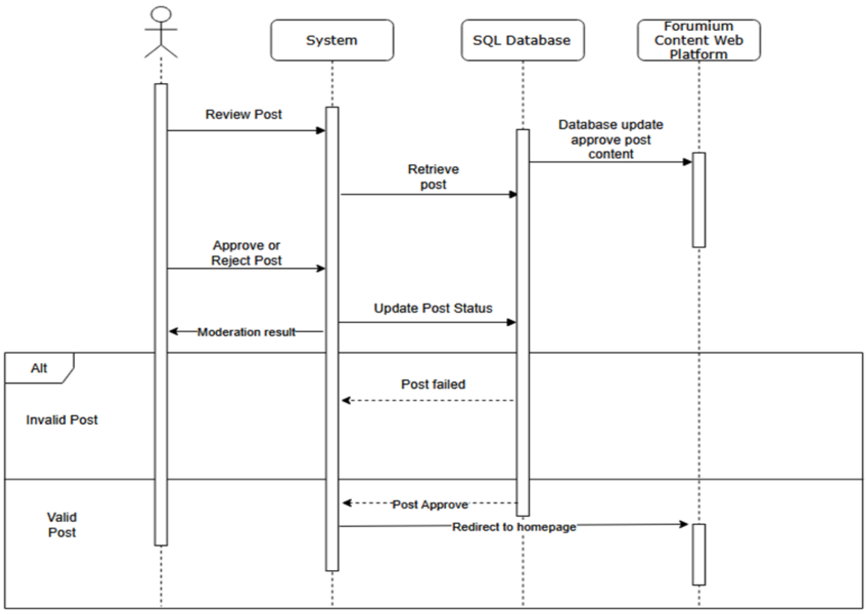
Sequence diagram of registration



Sequence diagram of login



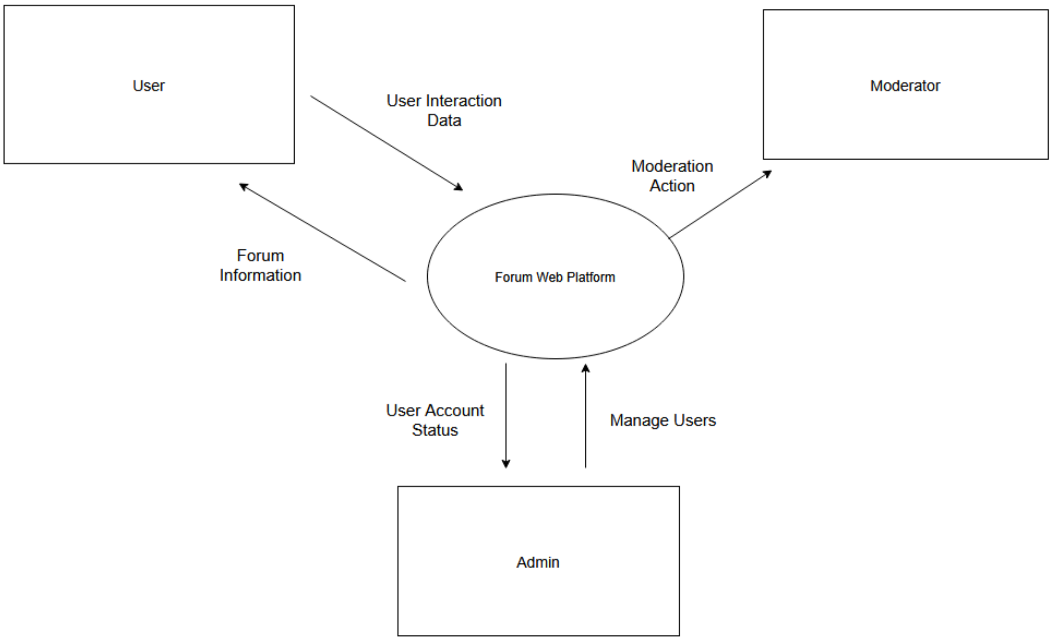
Sequence diagram of creating a forum post



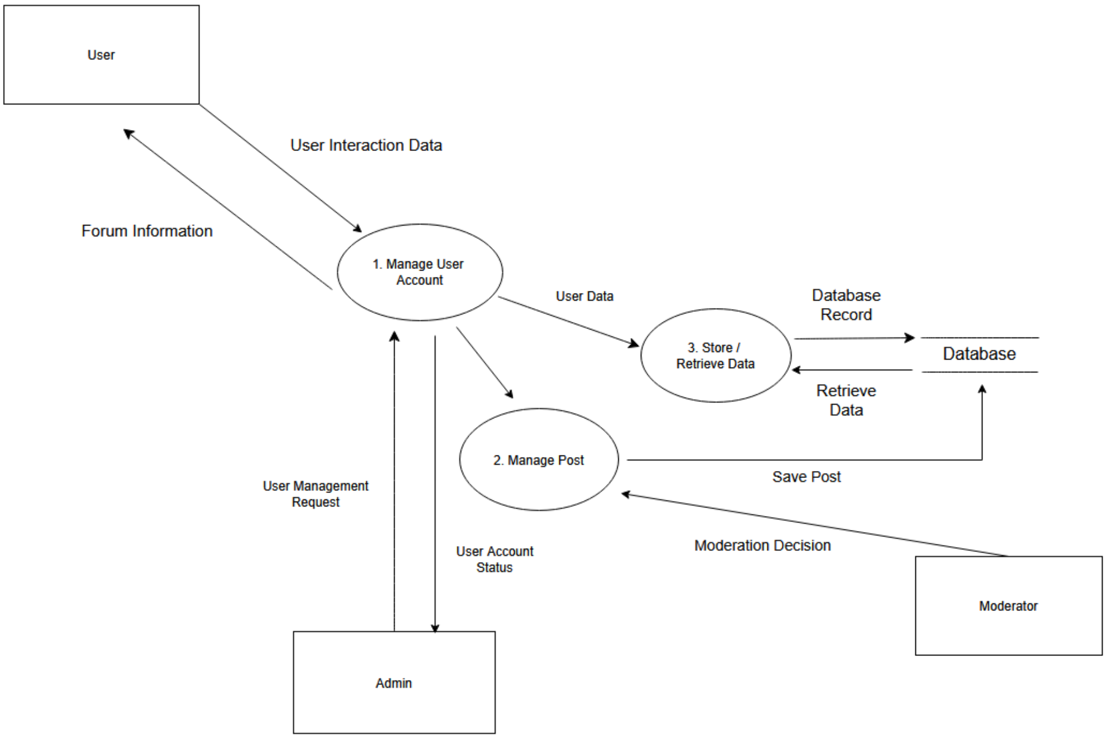
Sequence diagram of moderating forum post

Analysis Models

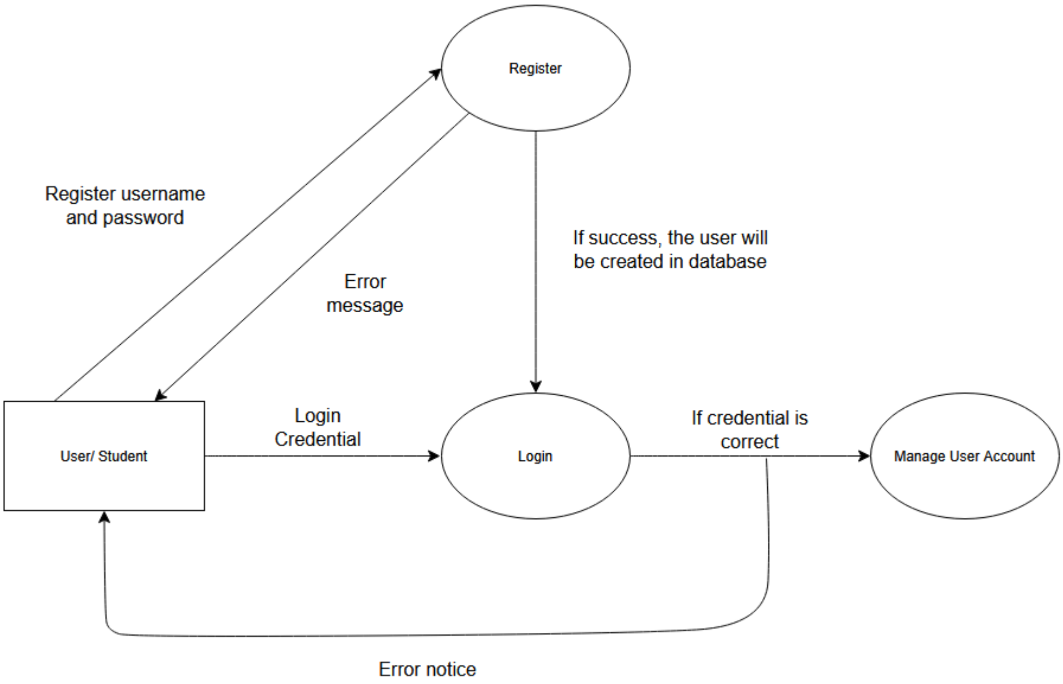
Data Flow Diagrams (DFD)



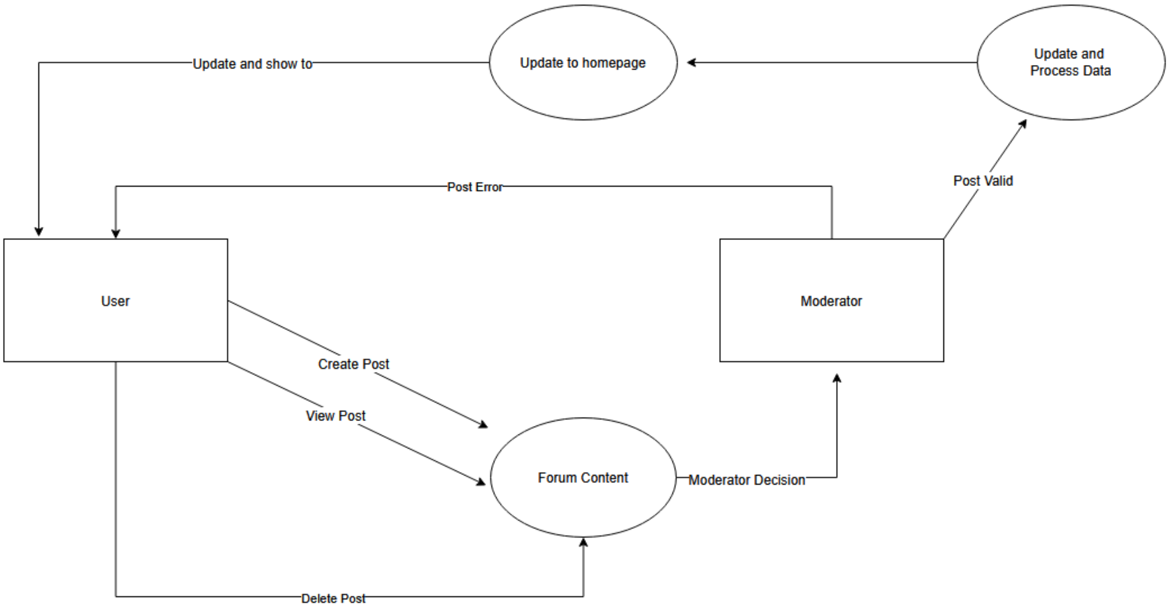
Level 0 - DFD



Level 1 - DFD

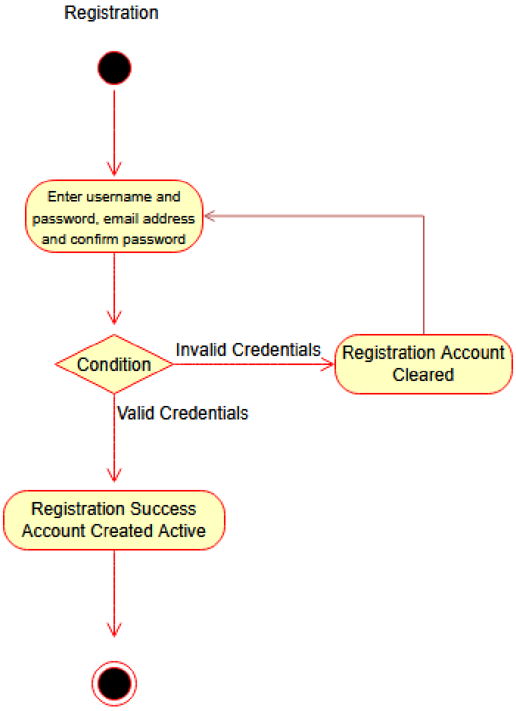


Level 1 – DFD for registration/login (extra)

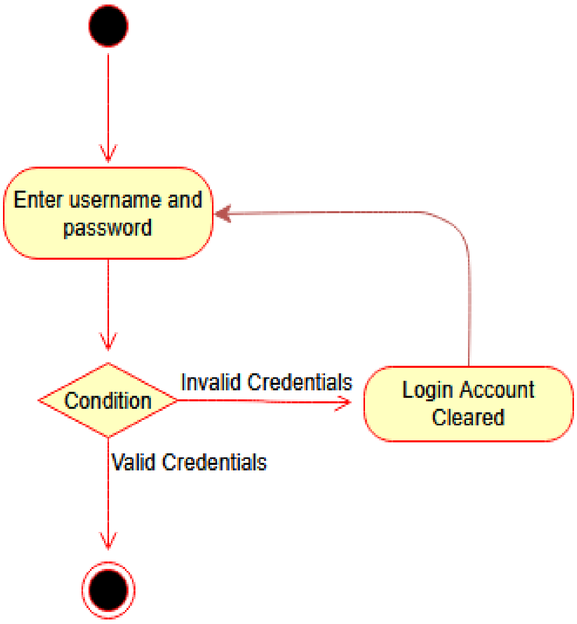


Level 1 – DFD for creating, viewing and delete forum post content (extra)

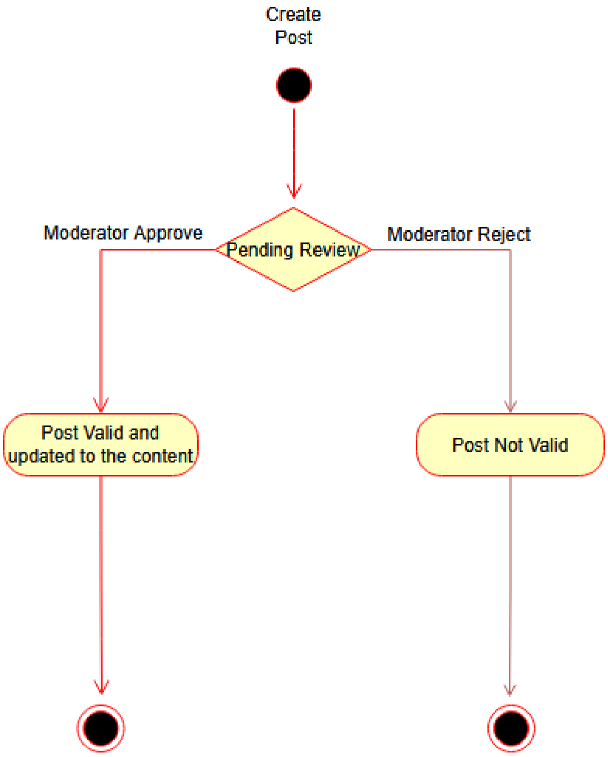
State Flow Diagrams (STD)


Registration
Enter username and password, email address and confirm password
Condition
Invalid Credentials
Registration Account Cleared
Valid Credentials
Registration Success Account Created Active

UML diagram for registration



UML diagram for login



UML diagram for user created post lifecycle

8.0 Specific Requirements

The web-based forum application is developed to provide an efficient platform for student communication and discussion within a university environment. The system must include features that allow users to create, view, edit, and delete forum posts across various discussion categories. The application should offer a clear and user-friendly interface to ensure users can participate in discussions without confusion.

In addition, the system enables university personnel such as admins and moderators to manage, monitor, and organize forum activities effectively. This includes moderating content, managing user roles, and overseeing discussion engagement. The system supports efficient resource utilization by reducing the need for manual communication and enabling streamlined online interaction within the university community.

8.1 External Interface Requirements

8.1.1 User Interfaces

1. The user interface for this web application is simple, easy to use and efficient.

2. The user interface should minimize the number of clicks to make each transaction. Any unnecessary pages are not tolerant and should be removed.

3. The user interface should minimize all the animation and fancy effects to reduce the probability to "hang" for low spec devices

8.1.2 Hardware Interfaces

1. To utilize this web application, Internet access will be required. Hence, there should be a connection between the user and the Internet.

8.1.3 Software Interfaces

1. Database should be consistent with records and data for every transaction update.

2. Databases should be updated from time to time.

3. Only administrators are allowed to access into the database and alter the setting or remove the setting.(Full access)

4. Forum web application shall communicate to the database to save user account, forum post, admin, and moderators details.

5. Users with any OS should be able to access the application unless his/her devices do not support internet access and browsers.

8.1.4 Communication Interfaces

1. This project supports all types of web browsers.

9.0 Non-Functional Requirements

9.1 Performance

1. Forum web applications by Flarem need to be hosted on the internet to serve and provide its service. Since this application is a simple program, therefore, most devices available today no matter its low end, mid end or high-end device should be able to run this application smoothly and perfectly

9.2 Reliability

1. The system must be free of any kind of bugs and inaccurate results

2. The application should work properly and consistent for any users

9.3 Availability

1. The application should be available in any device with internet connection and web browser

2. The application should be available on the DNS 24/7. To achieve this, the contractual agreement should be done with ISP so the system will ensure high availability on the internet.

9.4 Security

1. When a user closes the browser, the user will automatically log out.

2. The login page will not display the user’s password when he/she is typing.

3. Only admins are allowed to alter the users and moderators' entities.

4. The database shall be encrypted.

9.5 Maintainability

1. After the application is created and before/after release, the app will constantly release updates to improve and maintain the user satisfaction or user experiences

towards this application. These updates will mostly depend on the feedback and comment by constantly doing surveys (refer to Software Quality Assurance Document).

9.6 Portability

1. The application should be able to be accessed by any means such as tablets, smartphones or laptops. It is because this application is just a web application.

Meeting Minute with University

Two meetings were held to discuss and gather the requirements of the system. During the first meeting, both parties restate their original intent on the project. The second meeting discussed the business operation and problems faced by the client.

First Meeting		
10 November 2025	11:00 AM – 12:00 PM	Google Meet
Type of meeting	Get together meeting	
Note taker	Luqman Eman bin Aiman (University) Akmal Danish (Flarem)	
Attendees	(University) Muhammad Abu bin Ali – Director of University Wan Nabil bin Abdullah – Operational Manager Luqman Eman bin Aiman – University Main Representative (Flarem) Faris Amsyari – Project Manager Syafiq Haikal – <need role> Muaz – <need role> Akmal Danish – Lead Programmer	
Forum Web Application Project Proposal Review		
30 Minutes	Akmal Danish	
Discussion	1. Presentation of Project Proposal by Flarem to University 2. System Diagram presentation of the web application 3. General presentation on roles and responsibilities of Flarem	
Conclusions	1. Flarem will build a Forum Web Application for University. 2. The Proposal is recognized by University. 3. Both parties will clearly communicate the requirements. 4. To achieve 3, another meeting will be held on identifying University strategy and the system analysis and design.	
2) University Tender Review		
30 Minutes	Luqman Eman bin Aiman	

Discussion	1. Presentation of Tender by University 2. Key problem areas are defined	
Conclusions	1. Flarem provided acknowledgement on coming up with a System Requirement Document. 2. Meeting on gathering requirement scheduled on 12 November, time 2:00 PM to 4:00 PM	
Action Items	Person Responsible	Deadline
Meeting Notice to be sent to all through email	Faris Amsyari	11 November 2025
Second Meeting		
12 November 2025	2:00 PM – 4:00 PM	Google Meet
Type of meeting	Gather detailed requirements meeting	
Note taker	Luqman Eman bin Aiman (University) Akmal Danish (Flarem)	
Attendees	(University) Muhammad Abu bin Ali – Director of University Wan Nabil bin Abdullah – Operational Manager Luqman Eman bin Aiman – University Main Representative (Flarem) Faris Amsyari – Project Manager Syafiq Haikal – Software Quality Assurance (SQA) Mir Mohammad Muaz – SCM Akmal Danish – Lead Programmer	
Welcoming and Introduction – 30 minutes		
	1) Faris Amsyari welcomed all members of the meeting. 2) Restate of meeting purpose – to gather requirements. 3) Muhammad Abu talked about the University strategic goals	
Requirement Gathering Meeting		
1.5 hours	Moderator: Akmal Danish	
Discussion	1) Wan Nabil provided information on the current operation 2) Operation and problem faced by University a. The university currently uses multiple channels (email, messaging apps, in-person communication) for academic and non-academic discussions. b. Managing communication across these disconnected platforms has become inefficient as student numbers increase. c. Informal platforms used by students lack proper organization, moderation, and record-keeping. d. Important announcements and discussions may be missed	

	or duplicated due to the lack of a centralized system. e. University management has identified the need for a centralized web-based forum to improve communication and information access.	
Conclusions	1) The university requires a centralized and structured platform to manage student communication and discussions effectively, as existing methods are fragmented and inefficient. 2) Therefore, there is a need to introduce a digital, web-based forum solution to support organized interaction and information sharing. 3) Flarem will prepare the System Requirements Specification document in accordance with the agreed project backlog.	
Action Items	Person Responsible	Deadline
System Requirement Document	Flarem	12 December 2025

flarem plus.

SOFTWARE DESIGN DOCUMENT

Introduction to Software Design Specification (SDS)

Flarem submitted its comprehensive proposal to the institutional tender regarding the development of a streamlined communication solution for modern student engagement. The proposal was formally accepted during Phase 1 of the project on 12 November 2025. Under this agreement, Flarem will develop Forumium, a dedicated web-based forum platform designed to centralize student discussions, peer-to-peer support, and administrative announcements within a secure and innovative digital environment. This project aims to transition the client from traditional, fragmented communication methods to a sophisticated technological ecosystem that prioritizes data integrity and community safety.

The current operation of student communication involves two major risks that hinder the efficiency of the academic environment. Firstly, there is a significant issue with information fragmentation and digital congestion, where vital academic updates are lost within unorganized social media groups and unofficial messaging channels. Secondly, the existing reliance on physical bulletin boards and manual inquiry systems creates unnecessary administrative bottlenecks and physical crowding at information desks. These legacy systems are no longer sufficient for the growing student population and the complexities of modern campus life.

With the rapid digital transformation of the education sector, the institution has observed a great influx of digital traffic as well as increased demand for real-time interaction. The top management of the organization strives to create a safe and organized experience for the students, ensuring that the exchange of information is both controlled and accessible. By implementing Forumium, Flarem will provide a structured web application that eliminates the chaos of unofficial channels, replacing them with a moderated, high-performance platform that fosters a secure intellectual community.

1.1 Forumium Platform Overview

Forumium is a web-based forum application designed to enhance the academic experience by providing a centralized digital space for students to collaborate, share resources, and engage in peer-to-peer learning. By moving academic discussions to a structured online environment, the platform eliminates the fragmentation of information found in casual messaging apps. Forumium offers a streamlined, accessible technological solution that fosters community engagement while allowing students to seek help and share knowledge asynchronously.

The software consists of:

1. **Web-based Interface:** A responsive frontend for seamless interaction across devices.
2. **Database Hosting Server:** A secure backend to manage user data, discussion threads, and media attachments.
3. **Account Management System:**
    - **Student Account:** To create discussion threads, reply to posts, and join specific academic interest groups.
    - **Moderator Account:** To manage content, flag inappropriate posts, and ensure community guidelines are upheld.
    - **Admin Account:** To oversee system health, manage user roles, and analyze platform engagement metrics.

1.2 Document Description

1.2.1 Introduction

1.2.1.1 Purpose

This **Software Design Document (SDD)** provides the development team with a comprehensive blueprint of the Forumium architecture to streamline the implementation phase and mitigate technical debt. It outlines the platform's user personas, hardware requirements, and the technical frameworks utilized in its construction. This document serves the following critical functions:

- To provide a functional and architectural perspective to the developers.
- To define the structural components of the forum and the logic behind post-threading.
- To serve as a benchmark for the assessment and evaluation of the final platform.
- To provide a framework for simulating user interactions during the testing phase.
- To verify that the platform meets the specific collaborative needs of the student body.
- To facilitate efficient maintenance and feature updates after the initial launch.

1.2.1.2 Scope

- This document provides high-level abstraction diagrams and data-flow schemas that are directly translatable into code.
- Detailed specifications regarding the hosting environment, database management systems, and frontend/backend frameworks are included.

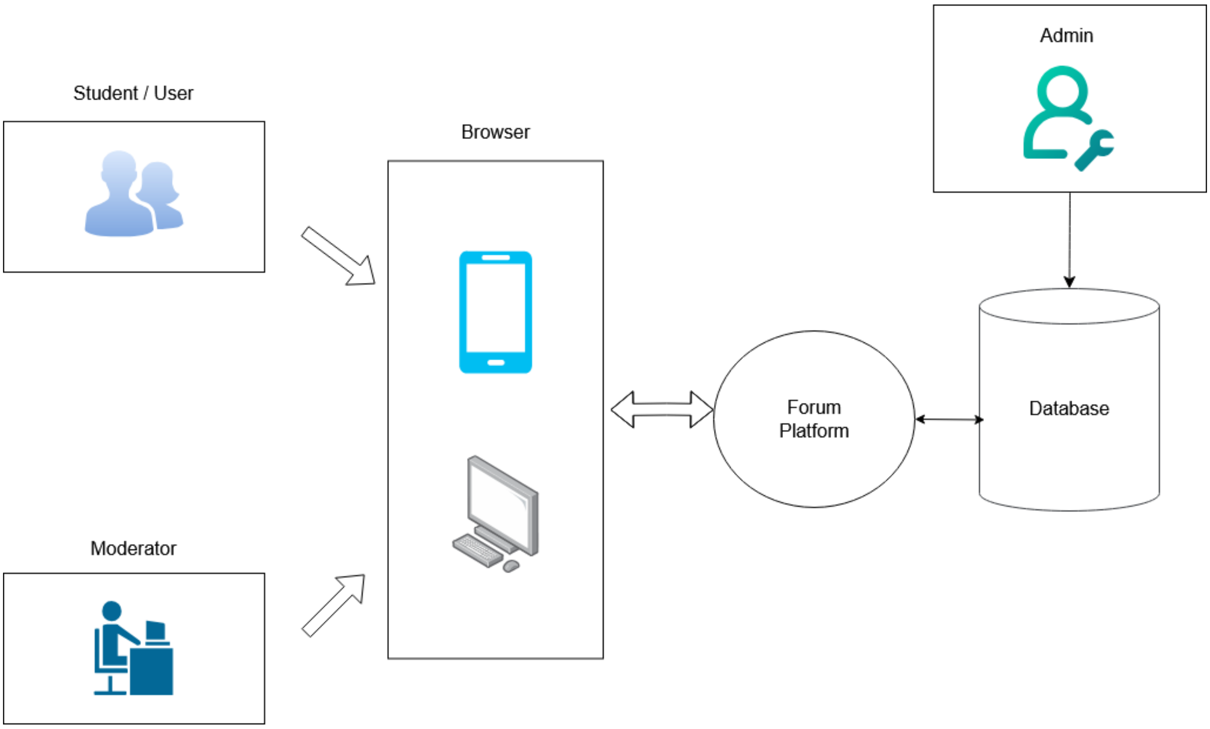
1.2.1.3 Intended Audience

This document is intended for the **Development Team** (Frontend, Backend, and Database Engineers), the **Quality Assurance (SQA) Team**, and the **User Experience (UX) Testing Team**.

System Overview

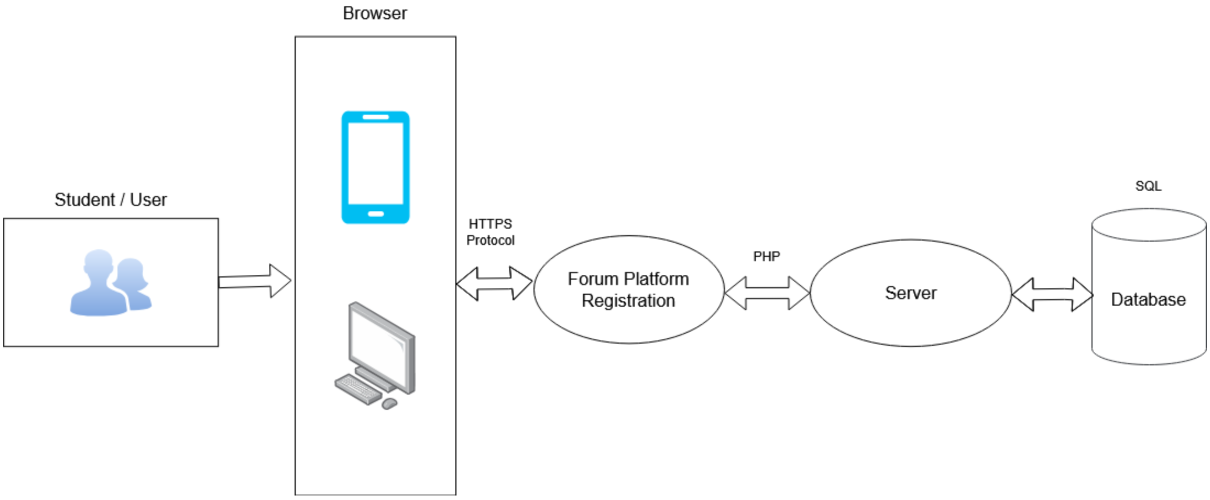
The basic design approach to the Forum Web Platform is by utilizing a browser base environment, student and users could access the website through their devices to post discussion, question and articles. These will then be stored in the database. The moderator will manage and moderate the post from users. The following figure gives an illustration of the roles involved for the system.

Forumium Web Platform System Diagram



User accounts can post their discussion to the platform. User post are moderated by moderator. The Admin account manages the user accounts and roles, and the database repository.

Web Technology Overview



The technology that is deployed to build the system are illustrated in the figure above. Users, who will access the system through a browser, will connect to the website via Hypertext Transfer.

Protocol Secure (HTTPS), which provides a secure connection to the internet. The website is served by mainly three languages: HyperText Markup Language (HTML), Cascading Style Sheets (CSS) and Javascript. These three languages provide a landing page for the user as well forming the basis of User-Interface/ User-Experience (UI/UX) of the website. PHP: Hypertext Protocol powers the website with flexibility and interactivity by being a scripting language that is invoked upon user's command. The Relational Database Management System (RDBMS) chosen is the MySQL database, which provides high scalability and high availability to the web app.

2.0 Design Considerations

2.1. Assumptions and Dependencies

The web application is assumed to run on any version of Microsoft Windows Server or any operating system like Linux which can operate a web server and support the execution of PHP language. Other than that, it is assumed that the user can utilise the web application using any electronic device such as a mobile phone or computer that supports an Internet connection and the user knows how to access and navigate the website. The biggest assumption is that the web application is planned to be extended to support a new feature based on the needs of our client. It is certain that additional functionality will be added to the proposed web application. In the matter of dependencies, the proposed application is dependent on the two servers to be capable of storing the required data. One of the servers is used for the data backup, in which there will be a domain that can be used by the clinic for the web interface by the potential customers. It is necessary for the web interface to have the right to access the database of the clinic's capacity at a particular time. Other than that, the web interface used by the customer should have a workstation with a stable Internet connection.

2.2. General Constraints

In terms of network connectivity, there is a constraint identified with this project. This constraint may exist due to the stability of the Internet network connection from the end-user. It can occur when there is a loss of Internet connection or server downtime during any active session from the user. To rectify this error, the users are required to log in and enter their password again before they can resume booking an appointment. Hence, during this time, an error message will appear on the screen depending on the occurrence of the error, such as “temporarily lost network connection” or “Server not found”.

2.3. Goals and Guidelines

The following goals must be addressed in the execution of the proposed web application.

2.3.1 Ease of Use

The system must be easy to use and provide a strong user experience. The users must be able to feel like they can navigate the web page and be able to comprehend how the system works to book their desired time slot for their appointment. As stated in the software requirement document, the number of clicks to make each transaction made by the user should be minimized.

2.3.2 Extensibility

The proposed system’s features must be extensible such as the new features and modifications on the application can be enabled as required by the users.

2.4. Development Methods

The system is designed by taking into consideration of several methods as listed below:

2.4.1 Reliability

To prevent any inaccuracy in the information provided by the system, regular checking will be done by the developers allow properly function application. This is also to ensure that the users will not be disrupted by any inconsistency.

2.4.2 Security

The users will be able to enter their information details safely without having to worry about data breaches, which will result in discrepancies in our product’s reputation. We aim to protect the data of the users by allowing only authorized persons such as the administrators to access the information.

2.4.3 Flexibility

The users must be able to access the web application using any electronic devices such as tablets, smartphones or even laptops. The system should be designed to have the same interface on any device and this is possible because the application is just a web application

2.4.4 Maintainability

Any future development such as modification and new features upon the request or feedback from the customers will be integrated into the system to ensure the satisfaction of the users and improve the user experience.

3. Architectural Strategies

This section describes the design strategies that affect the overall organization of the appointment web application system and its higher-level structures.

The system is created using CSS, PHP, JavaScript programming languages and attached to a database for information storage and retrieval purposes. The reasons for using the programming languages stated above is that it specifies the information contained in each item on the webpage, allows interactive features to be displayed on the webpage and defines the style for the web page.

Other than that, upon the user's request to add new functionalities and features, the system can be extended and enhanced further. This is important for our web application to stay relevant as well as improve customer's experience by launching new features. However, to ensure our target audience can use the new features seamlessly and without any interruption, it is important to test out the features first. In terms of hardware interface, the user is required to have an Internet connection when he/she is accessing the web platform. Meanwhile, for the software interface, it is essential for the database to be consistently updated for every transaction made by the users and for security purposes, only authorized users such as administrators are allowed to access the database to alter or remove the settings. Meanwhile, for error detection and recovery, a test case design will be produced to be used on the software. The SQA team will then use the test case to review, test and audit the software consistently. Audits can be scheduled or unscheduled.

4. System Architecture

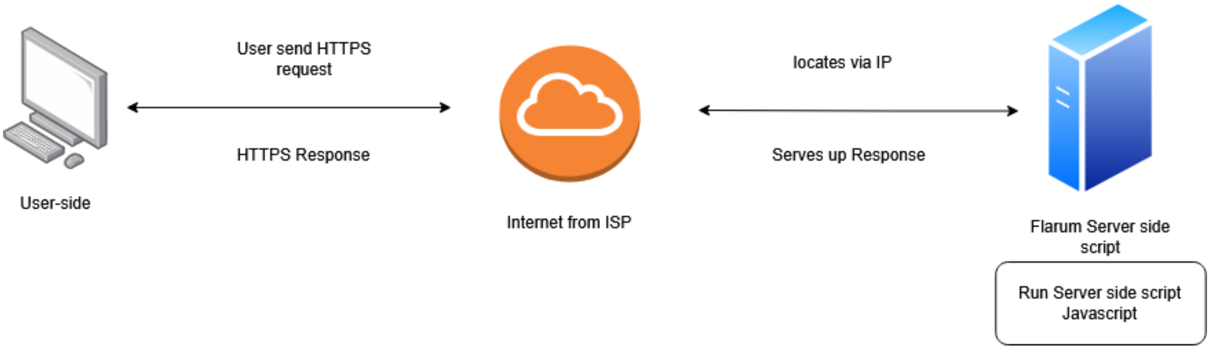
This section will provide a good level overview of the web app under development to list the major responsibilities of Flarem. Subsequently, the document provides a detailed component description in the form of a Data Flow Diagram. The primary intuition to this approach is to provide a general understanding of how the web app technology functions, how and why the system is composed as it is and how individual parts work together to create the holistic inter-dependence system.

4.1. Subsystem Architecture

4.1.1 Overview of Client/Server Model of Web App System Architecture

A particular concept important to web app development is the distinction between the client-side and server-side. The internet is largely utilizing this model to serve web users. User devices are connected via a network to a central server to request a service. Based on the request, the central server will locate the corresponding server and then direct the response to the end-user. In the case of web apps, client devices send HTTPS requests to their Internet Service Provider (ISP) server for the web pages and other applications. The server will locate Flarem local server through the Internet Protocol (IP) address and serve up the responses.

Client/Server Model of a Forumium Web Platform

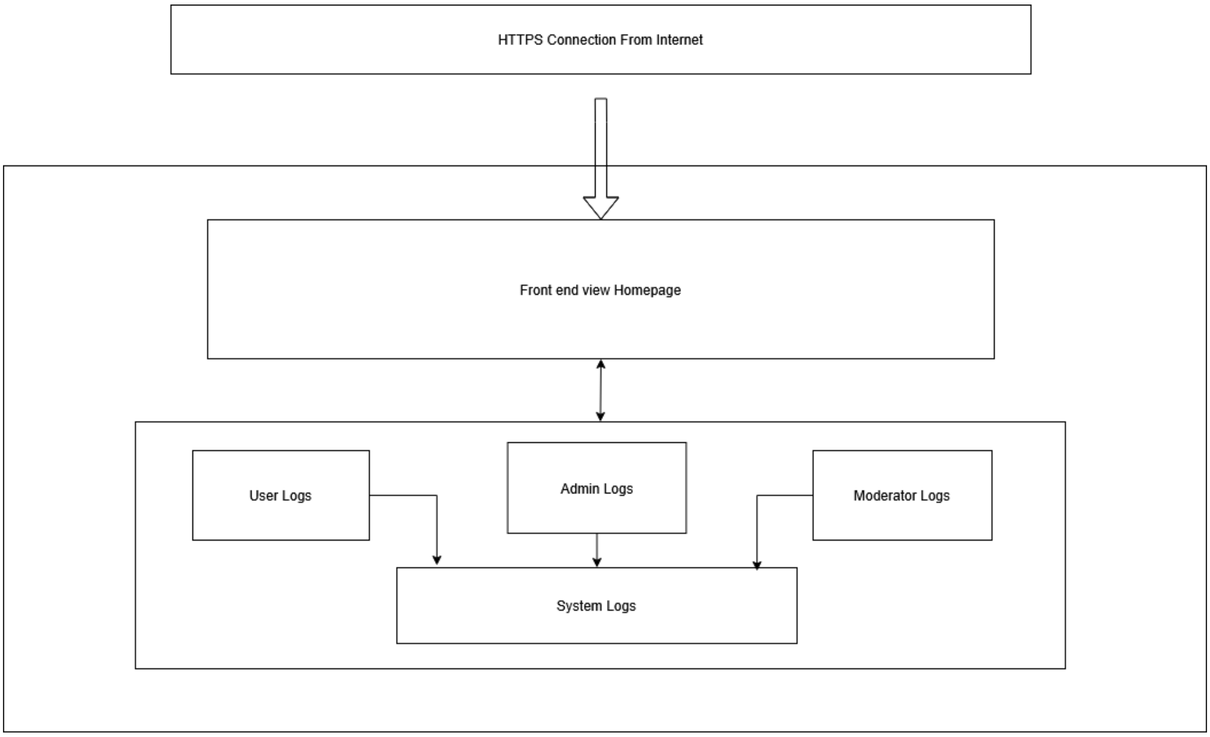


The interactivity and response-ability of the web pages comes from the power of scripting languages. The scripts running in runtime provide greater interactivity to the users than a static content written in HTML. The scripts running in client-side, after the web pages loaded are generally written in Javascript while in the server-side, the language used is PHP.

4.1.2 System Block Diagram of Flarem Local Server

The block diagram provides a simplified overview of Flarem's local server architecture to the web app in development.

System Block Diagram for Web Platform

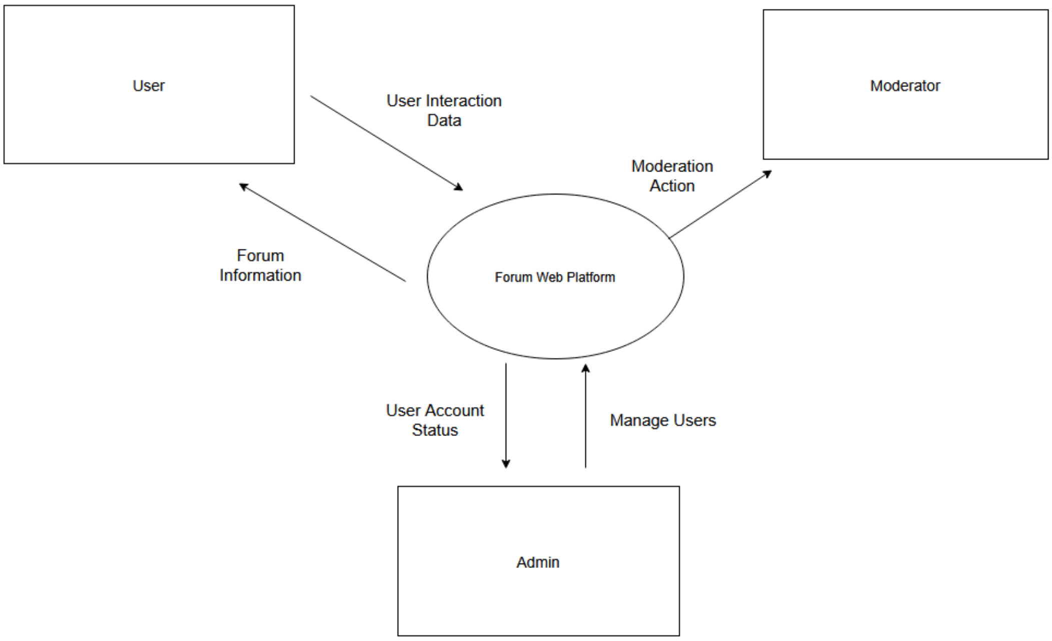


The server will receive HTTPS requests through the public IP address to serve the web page of Appointment Web Application. The PHP scripts on AlphaDevX local server will serve up the webpages. All actions on the server will create a log that is stored in the system log of the database. These log reports will be used for auditing purposes.

4.1.3 Data Flow Diagram (DFD)

4.1.3.1 Level 0 DFD

The context diagram, aka Level 0 DFD, is a single diagram representing the entire system. The subsequent levels of DFD will explore in detail how each of the processes handles the data flow. The diagram will continue to expand until each "bubble" refers to only a single, simple and executable process.

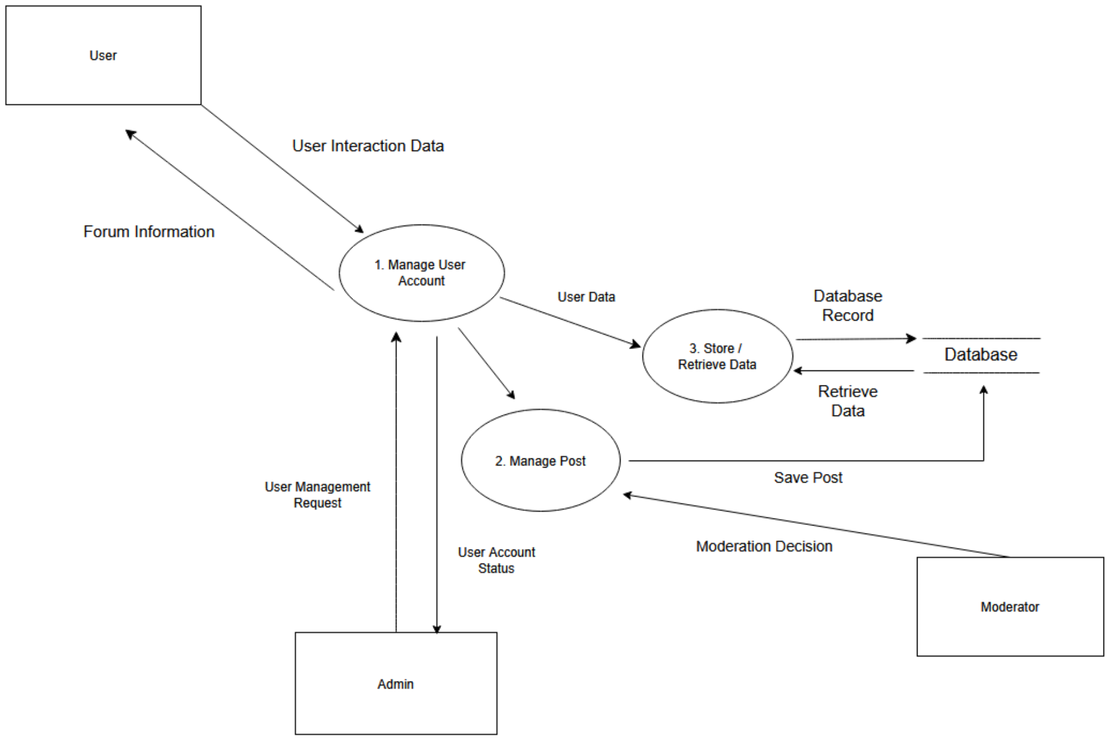


Level 0 DFD - Three external entities

Diagram illustrates three external entities interacting with the **Forumium Web Platform**, namely the **User**, **Moderator**, and **Admin**.

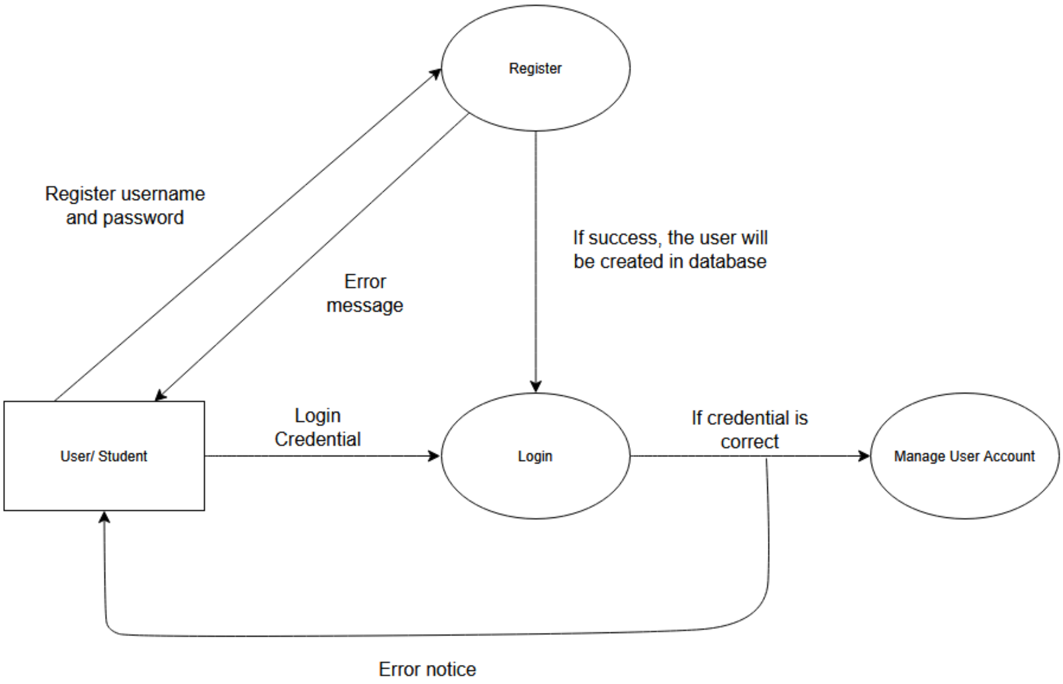
Users interact with the system to manage their accounts and participate in forum activities such as viewing and creating posts. The system processes these interactions and provides relevant forum information to the users. Moderator interacts with the platform to review and moderate user-generated content to ensure compliance with forum guidelines. Admin is responsible for managing users and overseeing system-related operations. The platform provides user account status and system-related information to the Admin as required.

4.1.3.2 Level 1 DFD

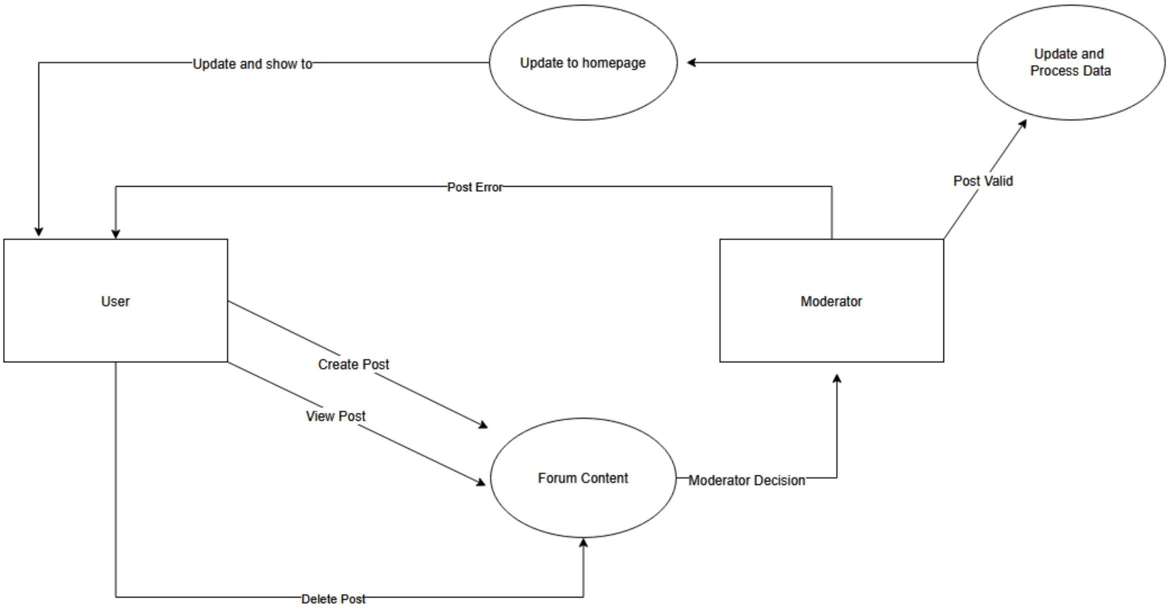


Level 1 DFD – Forumium Web Platform

This Level 1 DFD depicts a student forum platform where Process 1 (Manage User Account) serves as the central hub for student interactions and administrative oversight, handling profile data and account status updates. Process 2 (Manage Post) facilitates content moderation by processing decisions from Moderators to maintain forum standards, while Process 3 (Store/Retrieve Data) acts as the system's data handler, managing the read/write operations for User Data and content between the functional processes and the central Database. Together, these components ensure that student interactions are authenticated, content is moderated, and all forum activity is persistently stored for reliable retrieval.



Level 1 - DFD for registration/login



Level 1 - DFD for creating, viewing, delete forum post content

5.0 Data Design

5.1 Data Dictionary

User Table

Attribute	Data Type	Size	Key	Description
user_id	int	-	PK	Unique identifier for each user
username	varchar	50	-	User’s display name
email	varchar	100	-	User’s email address
password_hash	varchar	255	-	Encrypted user password
role	enum	-	-	User role (User, Moderator, Admin)
account_status	enum	-	-	Account status (Active, Suspended)
created_at	datetime	-	-	Account creation timestamp

Name	Type	Default	Example
user_id	int (pk)	Auto Increment	101
username	varchar(50)	NULL	arashi
email	varchar(100)	NULL	malish@mail.com
password_hash	varchar(255)	NULL	$2b$12$X9
role	enum(user, moderator, admin)	User	User
account_status	enum(active, suspended)	Active	Active

created_at	datetime	CURRENT_TIMESTAMP	2026-01-01 10:30:00

Post Table

Attribute	Data Type	Size	Key	Description
post_id	int	-	PK	Unique identifier for each post
user_id	int	-	FK	References User.user_id
title	varchar	150	-	Post title
content	text	-	-	Post content
post_type	enum	-	-	Discussion, Question, or Article
post_status	enum	-	-	Pending, Approved, Rejected
created_at	datetime	-	-	Post creation time

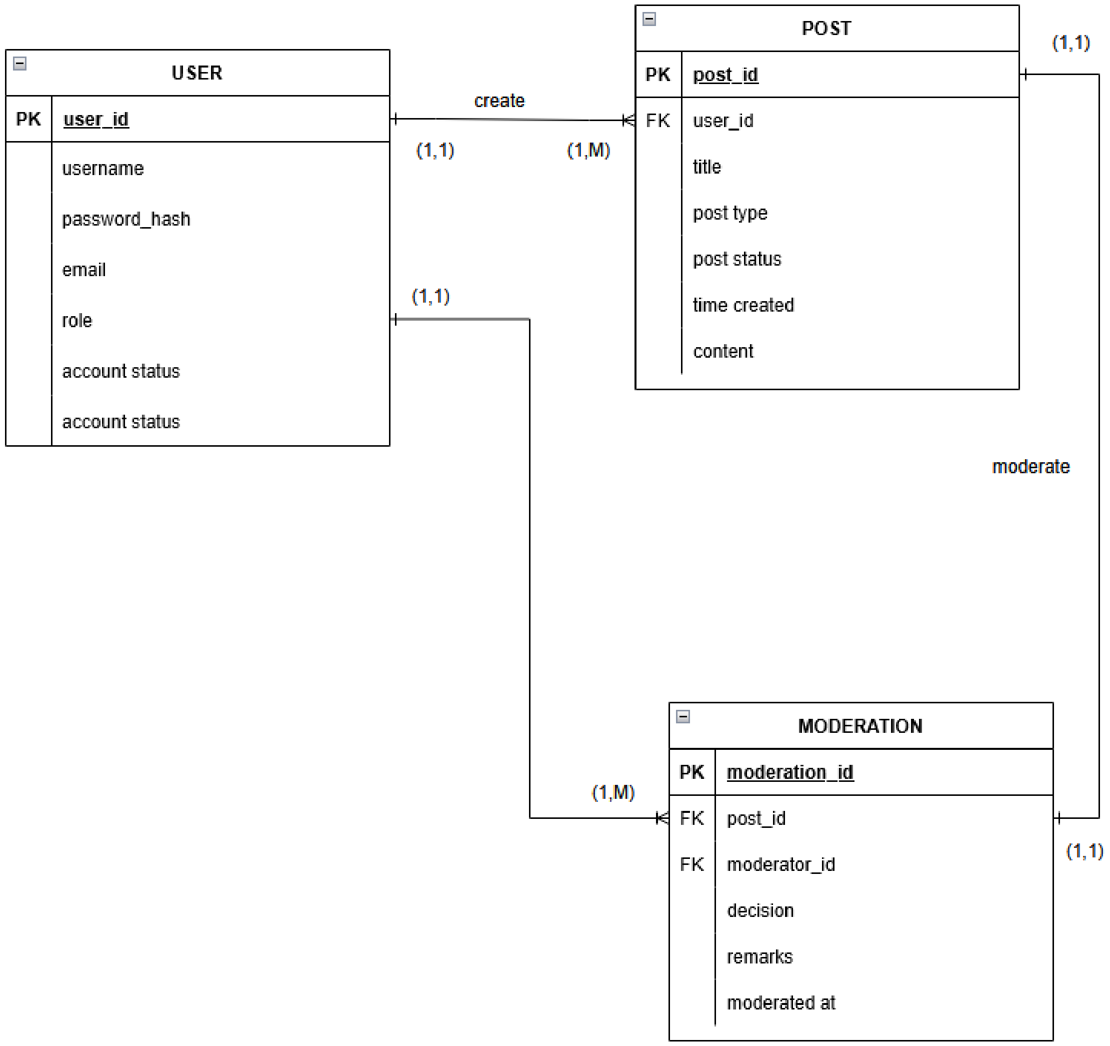
Name	Type	Default	Example
post_id	INT (PK)	Auto Increment	5501
user_id	INT (FK)	NULL	101
title	VARCHAR(150)	NULL	How to study electronics?
content	TEXT	NULL	Any tips for analog circuits?
post_type	ENUM(Discussion, Question, Article)	Discussion	Question
post_status	ENUM(Pending, Approved, Rejected)	Pending	Approved
created_at	DATETIME	CURRENT_TIMESTAMP	2026-01-01 11:05:00

Moderation Table

Attribute	Data Type	Size	Key	Description
moderation_id	int		PK	Unique moderation record
post_id	int		FK	References Post.post_id
moderator_id	int		FK	References User.user_id
decision	enum			Approve or Reject
remarks	varchar	255		Moderator remarks
moderated_at	datetime			Moderation timestamp

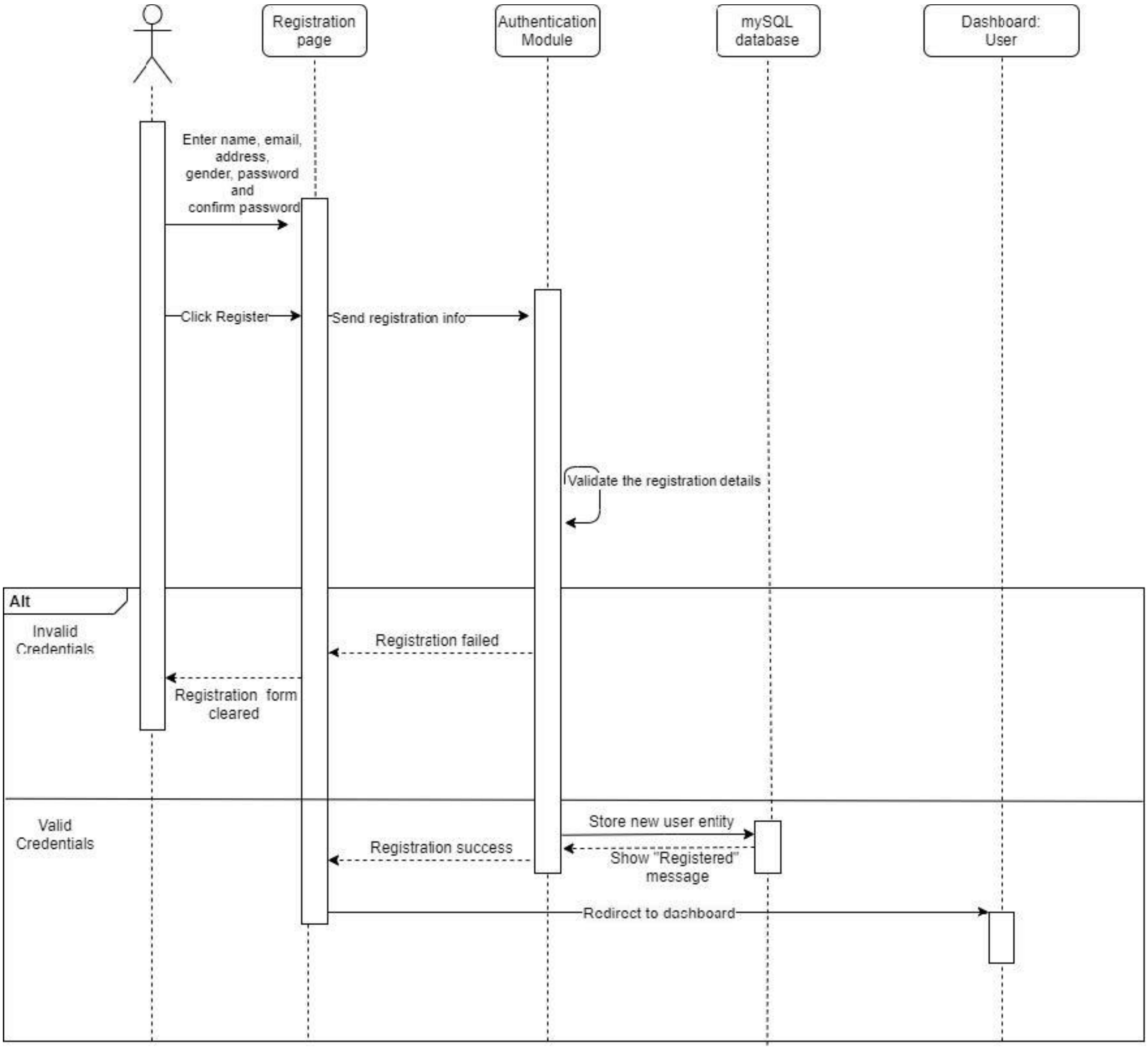
Name	Type	Default	Example
moderation_id	int (pk)	Auto Increment	9001
post_id	int (fk)	NULL	5501
moderator_id	int (fk)	NULL	205
decision	enum(approve, reject)	Approve	Approve
remarks	varchar(255)	NULL	Spam content
moderated_at	datetime	CURRENT_TIMESTAMP	2026-01-01 12:00:00

5.2 ERD Diagram

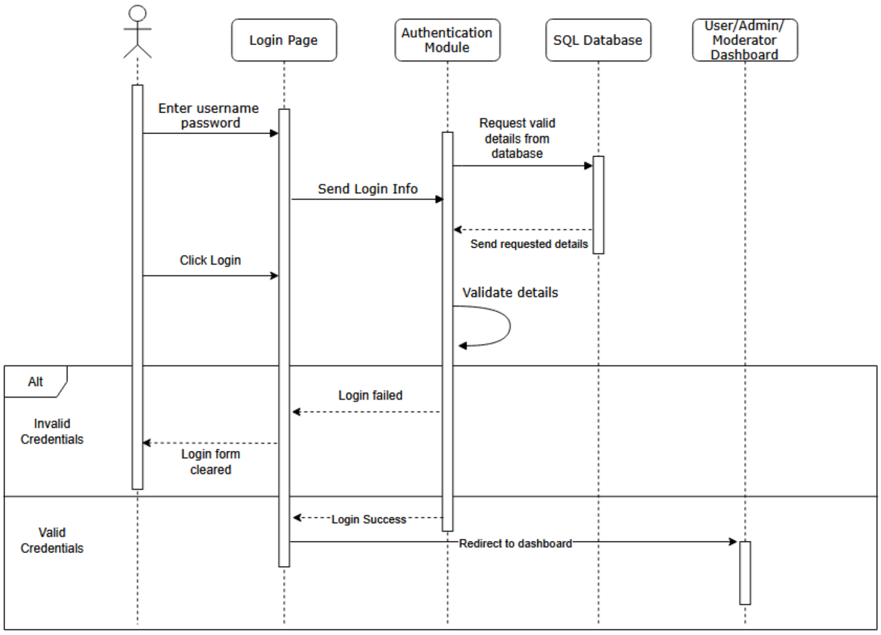


6.0 Component Design

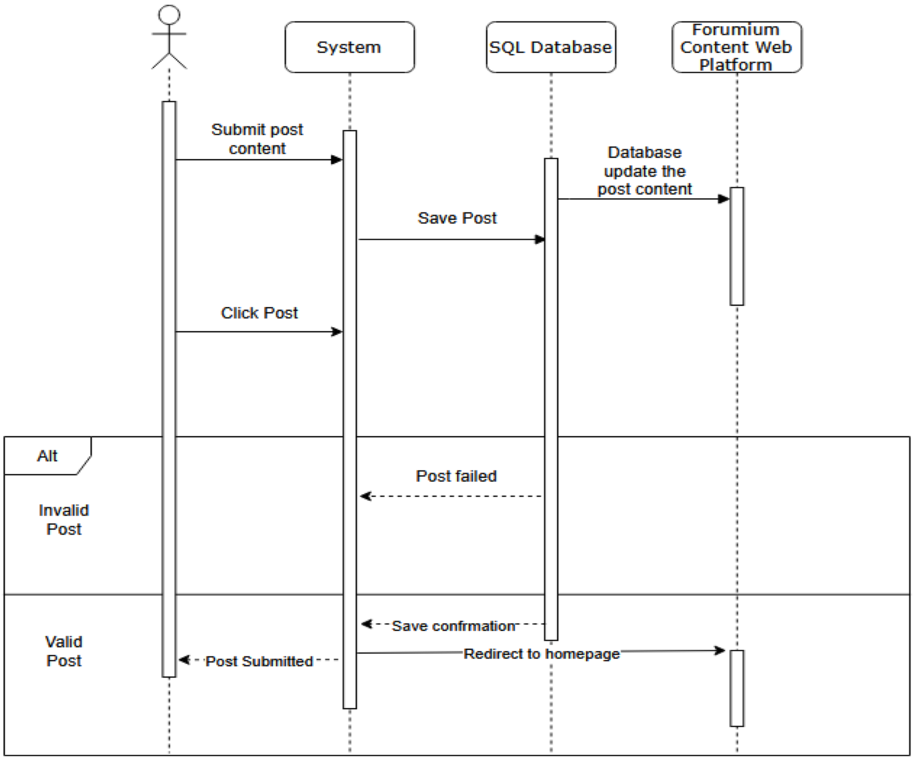
6.1 UML Sequence Diagram



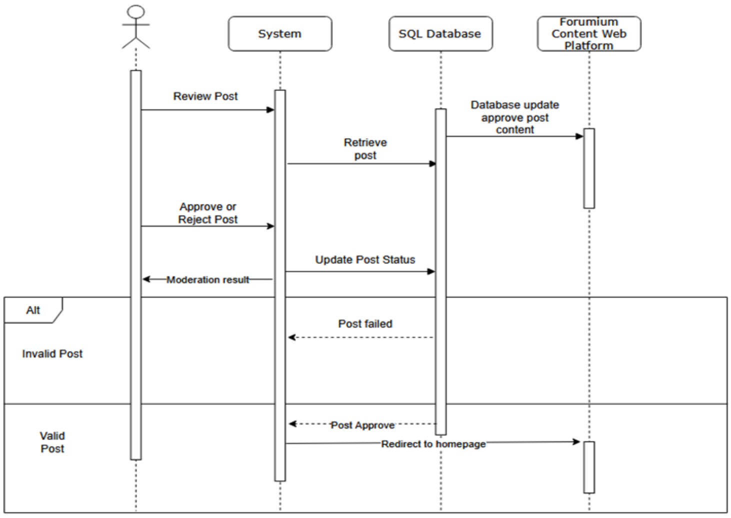
Sequence diagram for registration



Login Page

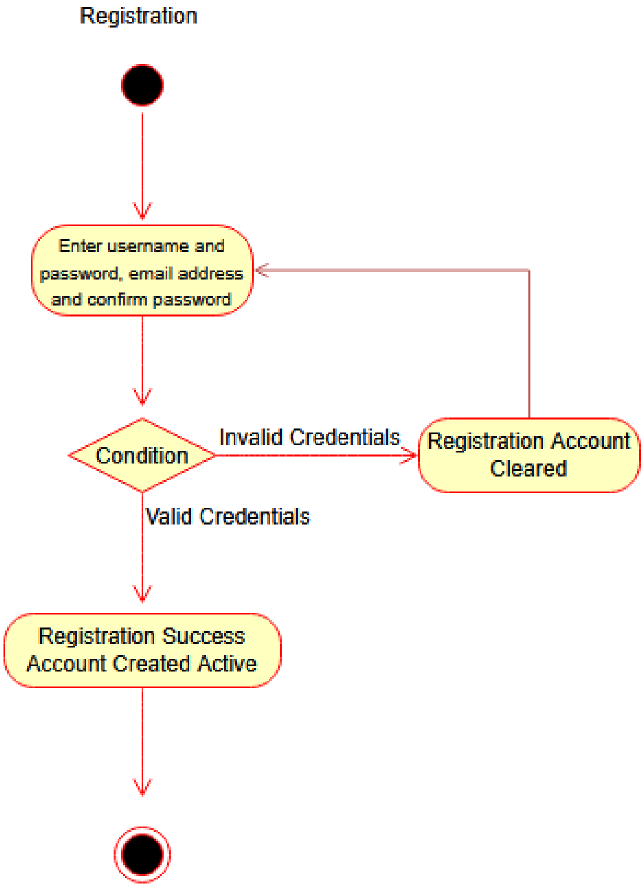


Create Post

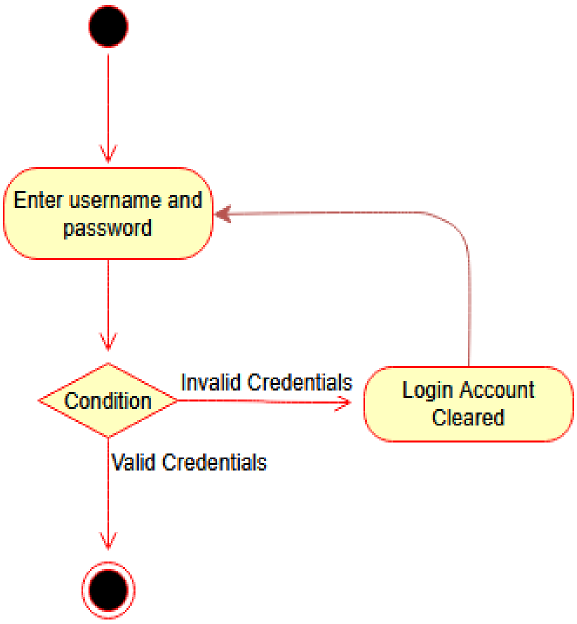


Moderate Post

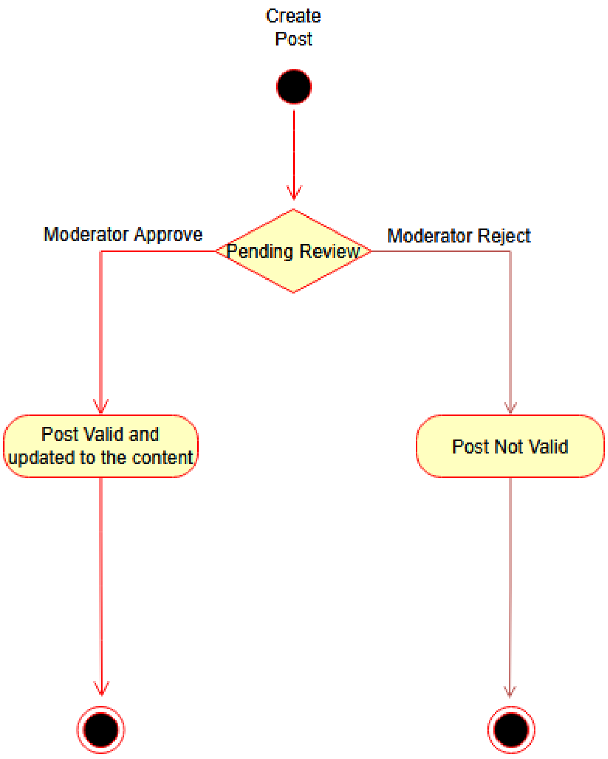
State-Transition Diagrams (STD)


Registration
Enter username and password, email address and confirm password
Condition
Invalid Credentials
Registration Account Cleared
Valid Credentials
Registration Success Account Created Active

UML diagram for registration

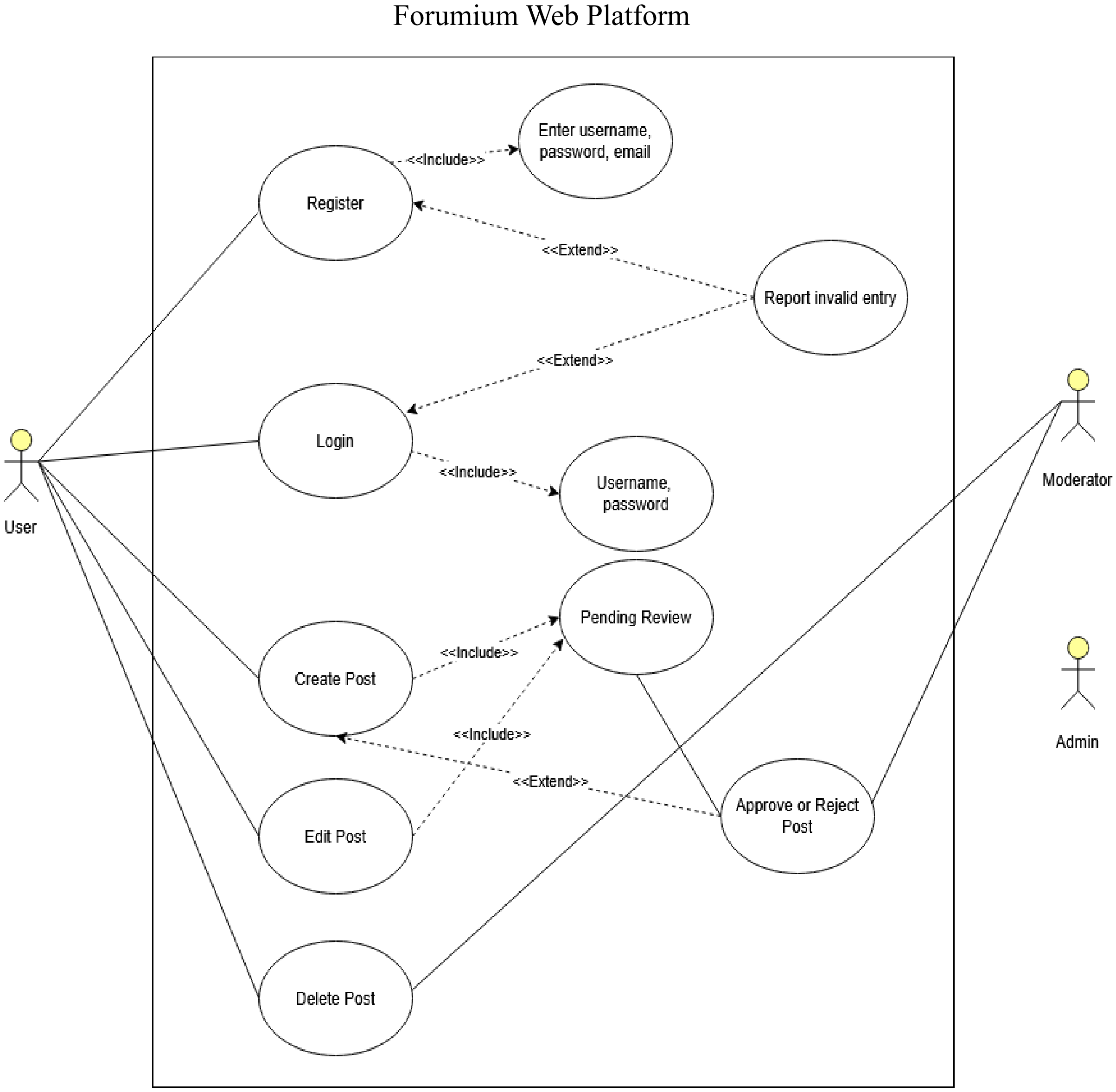


UML Diagram for login



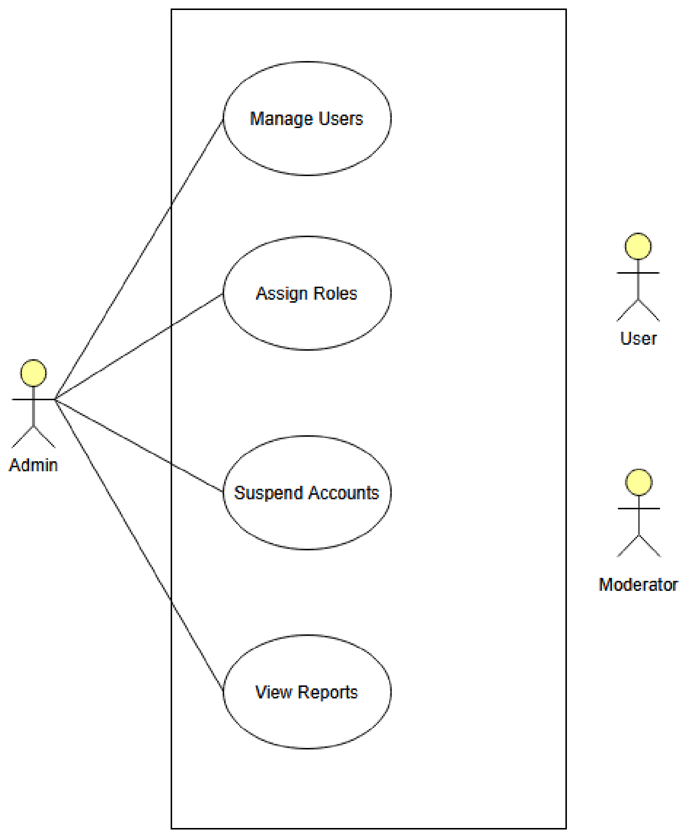
UML Diagram for User Created Post Lifecycle

7.1 Use Case Diagram



Use case diagram for user's point of view

Item	Description
Actors	User Moderator Admin
Main Success Scenario	1. The user accesses the Forum Web Platform and registers an account by entering a valid username, password, and email. After successful registration, the user logs into the system using valid credentials and gains access to the platform features. 2. The user then creates a post, which is automatically submitted for pending review. A moderator reviews the post and approves it, after which the system publishes the post and makes it visible to other users. The user may subsequently edit or delete their post as needed while using the platform.
Alternate Flows	1. If the user enters invalid or incomplete information during registration or login, the system reports an invalid entry and prompts the user to re-enter the required details. If a submitted post is rejected by the moderator, the system updates the post status to rejected and notifies the user, who may choose to edit and resubmit the post for another review. 2. At any point after creating a post, the user may also choose to delete the post, in which case the system removes it and terminates the posting process.



Use case diagram for admin's point of view

Item	Description
Actors	1. Admin 2. Manager 3. User
Main Success Scenario	1. The admin account must exist in the system and have valid administrative privileges. The Forum Web Platform and database must be operational, and the admin must be successfully authenticated before performing any administrative actions.

Alternate Flows	1. If the admin attempts to manage users or assign roles to a non-existent or invalid account, the system displays an error and prevents the operation. 2. If a role assignment or suspension action cannot be completed due to system errors or insufficient privileges, the system notifies the admin and no changes are applied. 3. When viewing reports, if report data is unavailable or a system failure occurs, the system informs the admin and allows the admin to retry or exit the function.
Pre-Condition	1. The admin account must exist in the system and have valid administrative privileges. The Forum Web Platform and database must be operational, and the admin must be successfully authenticated before performing any administrative actions.

8.0 Human Interface Design

A web platform's "initial impression" is crucial since a poorly designed website may deter and disappoint potential users. Therefore, when creating the UI/UX of the online application, the principles of human interface design should be monitored. Bruce Tognozz established a set of essential traits that any interface ought to have. The qualities are as follows:

8.1 Design Principles

1. Anticipation

It should appear as though the design is predicting what the user will do next. For instance, a site designer ought to provide the customer account group with a "book appointment" navigation feature.

2. Communication

The user interface (UI) should show the user's current state of a process or action, such as a text message indicating that a booking slot's approval is pending.

3. Consistency

The online application should maintain a similar UI layout, design, and functionality.

4. Restricted autonomy

Ensuring that there is a mechanism of authentication for access to critical data to prevent compromising data integrity.

5. Efficiency

The design should be simple to use and shouldn't interfere with the user's workflow or ability to perform tasks.

6. Adaptability

The web application can be accessed by the user either randomly or sequentially. A navigation bar could be used to do this.

7. Focus

The website's content should put the needs and actions of the user first. As a result, it must show the pertinent data in the main emphasis.

8. Fitt's law

According to Fitt's law, the ratio of the target's width to its distance determines how long it will take to go there quickly.

9. An object with a human interaction

A collection of reusable web design content already exists. You don't have to start from scratch.

10. Latency reduction

Users need not be concerned with the internal operation of the system. If they are required to wait, indicate it with some graphical content like rotating wheels, or a loading sign etc.

11. Learnability

The web should not have too steep a learning curve. It is functions should be clear cut and can be easily learnt.

12. Metaphors

Simulate other real events that have similar operational functions with the web. For example, recreating a physical notebook design for the appointment log system.

13. Maintain work product integrity

Whatever the user had keyed-in or entered a value or commanded should be recognize and stored in the system without

14. Readability

If there is text, it should be made readable no matter the age group of the user.

15. Track state

This design involves the use of web cookies to retain some of the browsing history of the user.

16. Visible navigation

The interface should function as if the user is always in the same place.