



Continuous Assessment for Laboratory / Assignment sessions

Academic Year 2023-24

Name: Amaranya Mishra

Course: Software Engineering

Year: T.Y. B.Tech.

SAP ID: 60004210210

Course Code: DJ19CEC601

Sem: VI

Batch: C31

Department: Computer Engineering

Performance Indicators (Any no. of Indicators) (Maximum 5 marks per indicator)	1	2	3	4	5	6	7	8	9	10	Σ	A vg	A 1	A 2	Σ	A vg
Course Outcome	1	2	2	2	3	2	5	4	4	6						
1. Knowledge (Factual/Conceptual/Procedural/ Metacognitive)	2	2	3	3	3	3	2	2	2	3		2	2	2		2
2. Describe (Factual/Conceptual/Procedural/ Metacognitive)	2	-	3	3	3	2	3	3	3	3		2	2	2		2
3. Demonstration (Factual/Conceptual/Procedural/ Metacognitive)	-	-	3	-	3	-	-	3	-	3		2	2	2		2
4. Strategy (Analyse & / or Evaluate) (Factual/Conceptual/ Procedural/Metacognitive)	2	3	3	3	2	2	3		3	3		2	2	2		2
5. Interpret/ Develop (Factual/Conceptual/ Procedural/Metacognitive)	-	2	3	2	-	3	3	3	-	-		-	-	-		-
6. Attitude towards learning (receiving, attending, responding, valuing, organizing, characterization by value)	3	3	2	3	2	3	-	-	3	3		2	2	2		2
7. Non-verbal communication skills/ Behaviour or Behavioural skills (motor skills, hand-eye coordination, gross body movements, finely coordinated body movements speech behaviours)	3	3	-	-	-	-	3	-	3	3		-	-	-		-
Total	12	13	12	14	13	13	14	11	14	14		10	10	10		10
Signature of the faculty member																

Outstanding (5), Excellent (4), Good (3), Fair (2), Needs Improvement (1)

Laboratory marks Σ Avg. = <u>13.5</u>	Assignment marks Σ Avg. = <u>10</u>	Total Term-work (25) = <u>23.5</u>
Laboratory Scaled to (15) =	Assignment Scaled to (10) =	Sign of the Student:

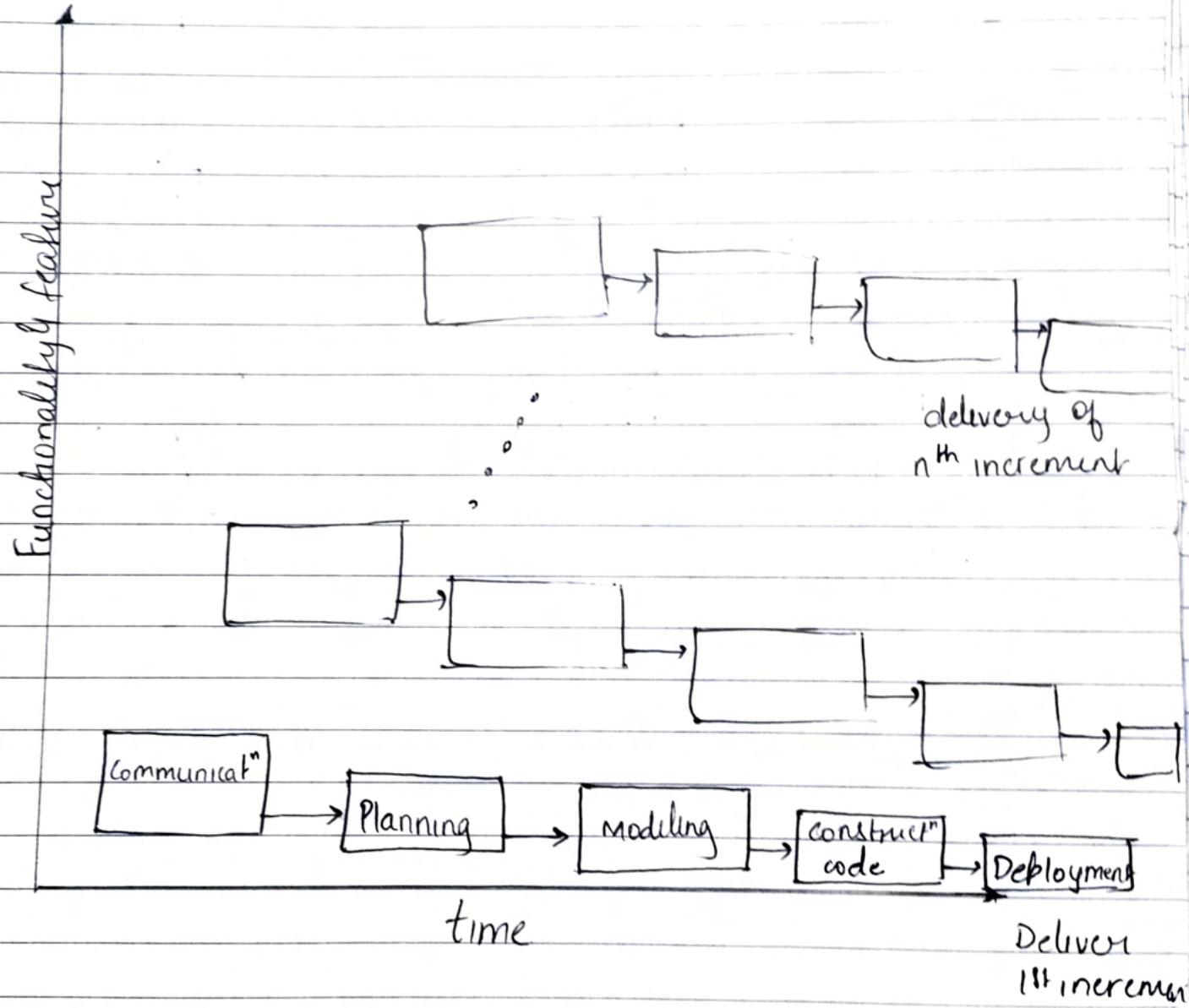
Signature of the Faculty member:
Name of the Faculty member:

30/11/24

Signature of Head of the Department
Date:

60004210210
Amaranya Mishra
COMPS - C31
SE lab

→ The Incremental Model



Agile Model → Extreme Programming

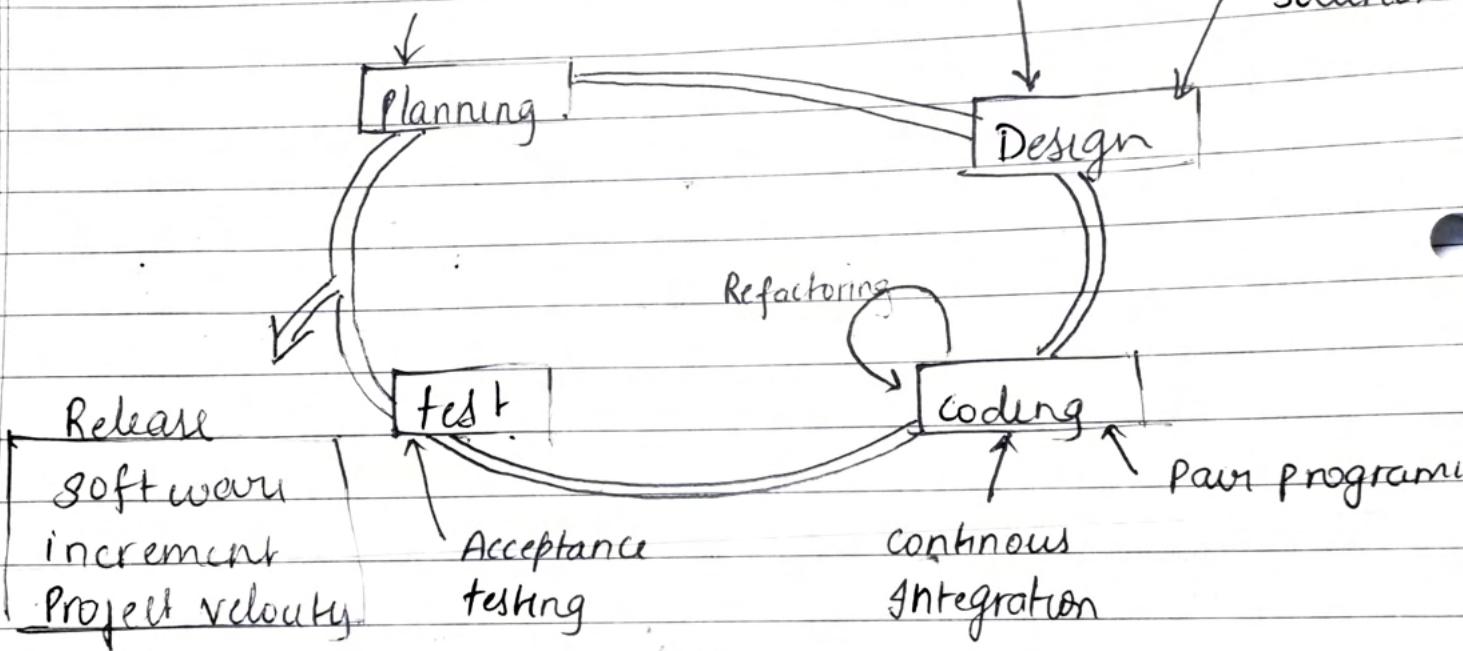
user stories

values

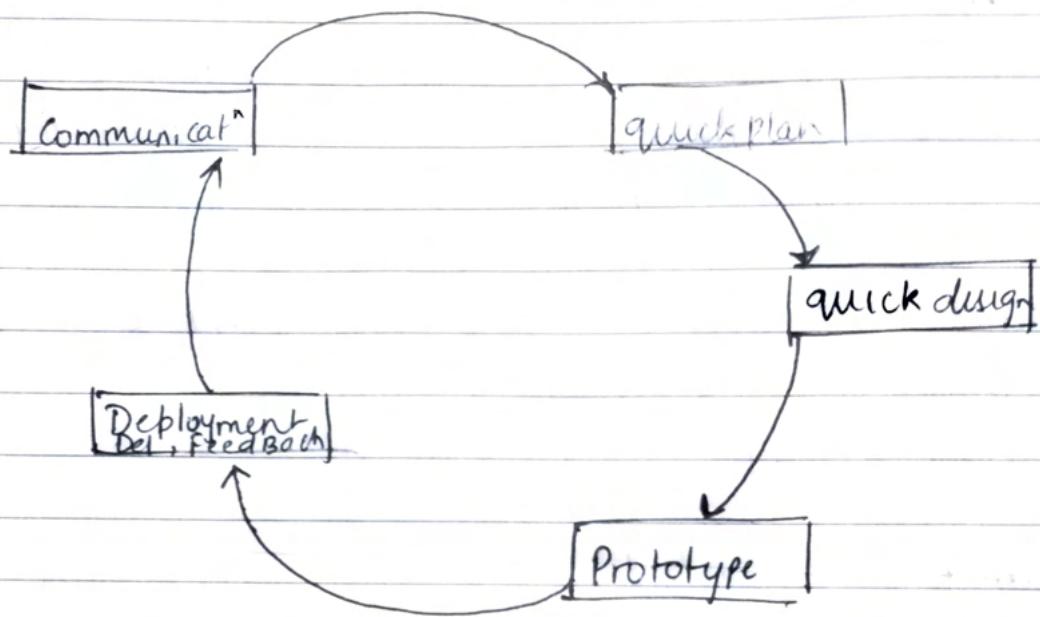
simple design
CRC card

spike

solution



→ Prototyping



Justification :

In context of our application chosen life cycle model is the Incremental Model. It is well suited for projects where initially requirements are unclear. It allows flexibility & adjustments throughout the development process.

Incremental model offers an adaptive approach to accommodate changes & enhancements as the circuitry application progresses.

Initial Increments : our case study involves the development of core features in circuitry such as seamless media sharing, user connections & basic collaborative functions

SE EXPERIMENT - 3

Aim: Identify Scenario & develop UML use case & class Diagram.

Theory :

Abstraction & Simplification :

class diagrams abstract complex systems into manageable components focusing on class & relationship ship.

Communication :

They provide a common language for stakeholders to understand & discuss the system structure & behavior.

Blueprint & implementation :

class diag. guide developers in writing code, defining class hierarchies & establishing relationships.

Analysis & design :

They are an early stage of analysis & design by identifying key classes & relationships facilitating iterative refining.

Modularity & reusability :

Organize classes into coherent units promotes modularity & reusability facilitating maintenance.

Documentation:

class diagram serves as a valuable document offering a visual representation of system's architecture for reference throughout the development life cycle.

Experiment 3

Darshit Sarda 60004210208

Amartya Mishra 60004210210

Gaurang Bhogle 60004210192

Shubham Mehta 60004210191

Aim: Identify scenarios & develop UML Use case and Class Diagram for the project

Theory:

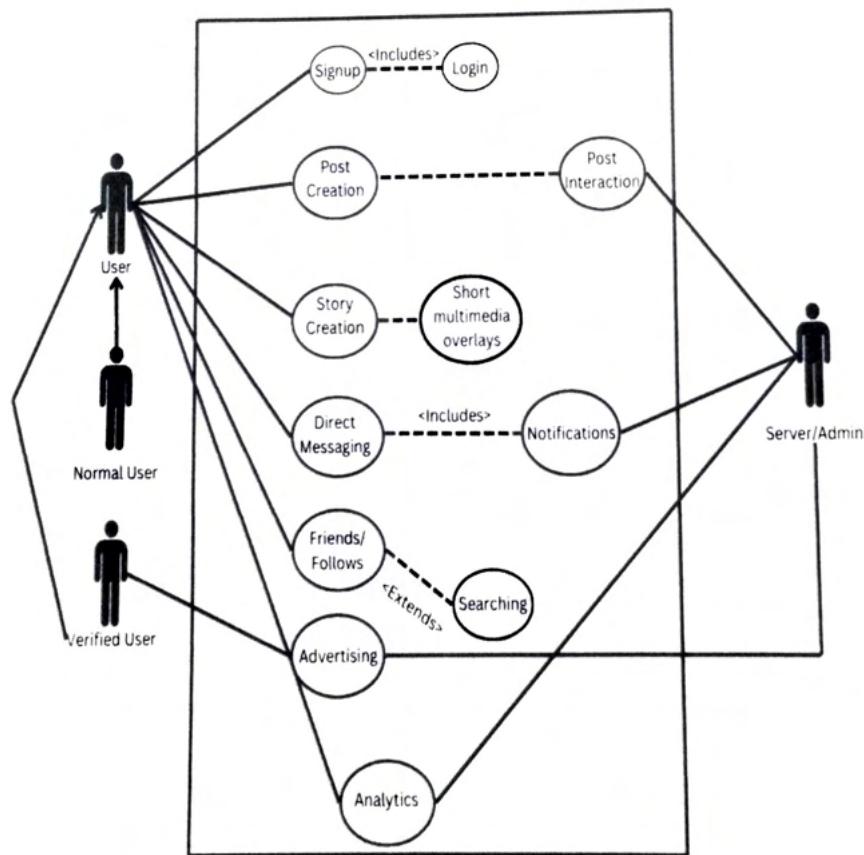
A class diagram in software engineering serves as a foundational tool for visualizing the structure and relationships within a system. It is a graphical representation of the classes, attributes, operations, and relationships among objects that make up the software system. Here's a breakdown of its importance and roles:

1. Abstraction and Simplification: Class diagrams help in abstracting complex systems into manageable components. By representing classes and their relationships visually, developers can focus on high-level design decisions without getting bogged down in implementation details.
2. Communication: Class diagrams serve as a common language between stakeholders, including developers, designers, and clients. They provide a clear and concise way to communicate the system's structure and behavior, facilitating collaboration and understanding among team members.
3. Blueprint for Implementation: Class diagrams provide a blueprint for the implementation phase of software development. Developers use them as a guide to write code, define class hierarchies, and establish relationships between objects.
4. Analysis and Design: During the early stages of software development, class diagrams aid in analysis and design. They help identify the key classes and their relationships, allowing developers to model the system's structure before writing any code. This iterative process enables refining the design based on feedback and changing requirements.
5. Modularity and Reusability: Class diagrams promote modularity and reusability by organizing classes into coherent units. Well-designed class hierarchies and relationships allow developers to encapsulate functionality within classes, making it easier to maintain and extend the system over time. Additionally, reusable classes can be leveraged across multiple projects, reducing development time and effort.
6. Documentation: Class diagrams serve as valuable documentation for the software system. They provide a visual representation of the system's architecture, which can be referenced by developers, testers, and other stakeholders throughout the software development process.

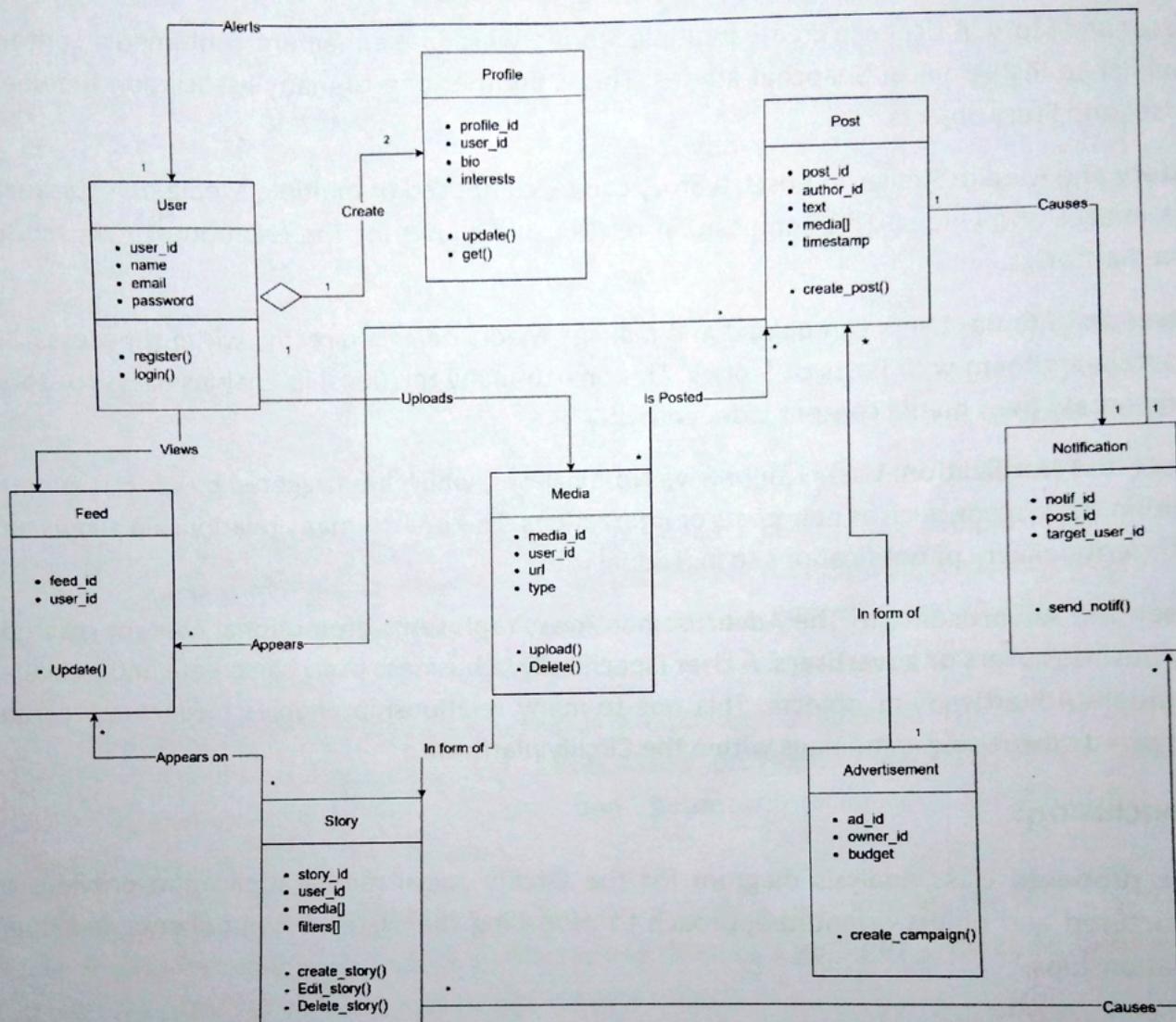
lifecycle. Additionally, class diagrams can be supplemented with additional documentation, such as class descriptions and usage guidelines, to further enhance understanding.

In essence, class diagrams play a crucial role in software engineering by facilitating communication, guiding implementation, promoting modularity and reusability, and serving as documentation for the system. They are indispensable tools for designing and understanding complex software systems, helping teams build robust and maintainable software products.

Use case Diagram:



Class Diagram:



User and Profile: The **User** class represents the core entity of the system, encompassing user accounts and authentication. Each **User** has an associated **Profile**, which stores additional information about the user, such as their bio and interests. This relationship is a composition, where a **Profile** object cannot exist without a **User** object.

User and Post: A **User** can create multiple **Post** objects, which can contain text, media, and other content. This relationship is a one-to-many association, where one **User** can be associated with multiple **Post** objects.

Post and Media: A **Post** can be composed of one or more **Media** objects, which represent images, videos, or other multimedia content. This is a composition relationship, where **Media** objects are owned by and exist within the context of a **Post**.

Feed and Post: The Feed class is an aggregation of multiple Post objects. A Feed represents the collection of posts that a User will see in their personalized content feed. This relationship allows for efficient retrieval and organization of posts for each user.

User and Story: A User can create multiple Stories, which are ephemeral multimedia content similar to Instagram or Snapchat stories. This is another one-to-many association between User and Story objects.

Story and Media: Similar to Posts, a Story can be composed of multiple Media objects, such as images and videos. This composition relationship allows for the creation of rich, multi-media stories.

User and Media: Users can upload and manage Media objects directly, without necessarily associating them with Posts or Stories. This one-to-many relationship enables users to store and access their media content independently.

User and Notification: Users can receive Notifications, which are triggered by various events within the system, such as new posts or interactions. This one-to-many relationship allows for efficient delivery of notifications to individual users.

User and Advertisement: The Advertisement class represents promotional content created by business users or advertisers. A User (specifically, a business user) can create and manage multiple Advertisement objects. This one-to-many relationship enables businesses to run targeted advertising campaigns within the Circlify platform.

Conclusion:

The proposed class analysis diagram for the Circlify social media application provides a structured and object-oriented approach to modelling the system's components and their relationships.

SE EXPERIMENT - 4

Aim: To develop activity diagram & DFD (upto 2 levels) for our case study

Theory:

Steps in activity Diagram are as follows:

- Start → The activity begins
- login or Signup : The user is prompted to either login with their existing account or sign up if they are new to the app.
- If New user : The user does not have pre-existing credentials , then they are directed to signup process They provide necessary information .
- Validated information : The user input is validated by the system to check if given details are acceptable or not
- Send verification mail : It is a part of security enhancing method as this step ensures only valid users get access to system functionalities.
- Verify Identity : The user checks Email & clicks on verification link to confirm their Identity.
- login : The user returns to app & continues with login by unputting credentials.
- login success full : The user is provided with auth. token & successfully can access the app.

→ Show loading page: The user is directed to landing page of circify app where they can view their feed, post content, interact with other users post & etc.

⇒ Swimlane diagram:

- User swimlane: It represents the actions & interactions performed by user such as login attempts etc.

- System swimlane: It represents actions performed by circify app. such as verifying credentials, storing information etc.

- Admin swimlane: It represents actions performed by admin on the system. This includes monitoring the data on platform, deleting & eliminating fraudulent accounts with misleading content.

Conclusion: Thus we created & represented the various elements of circify using Activity, swimlane & DFD diagrams.

Experiment 4

Darshit Sarda 60004210208

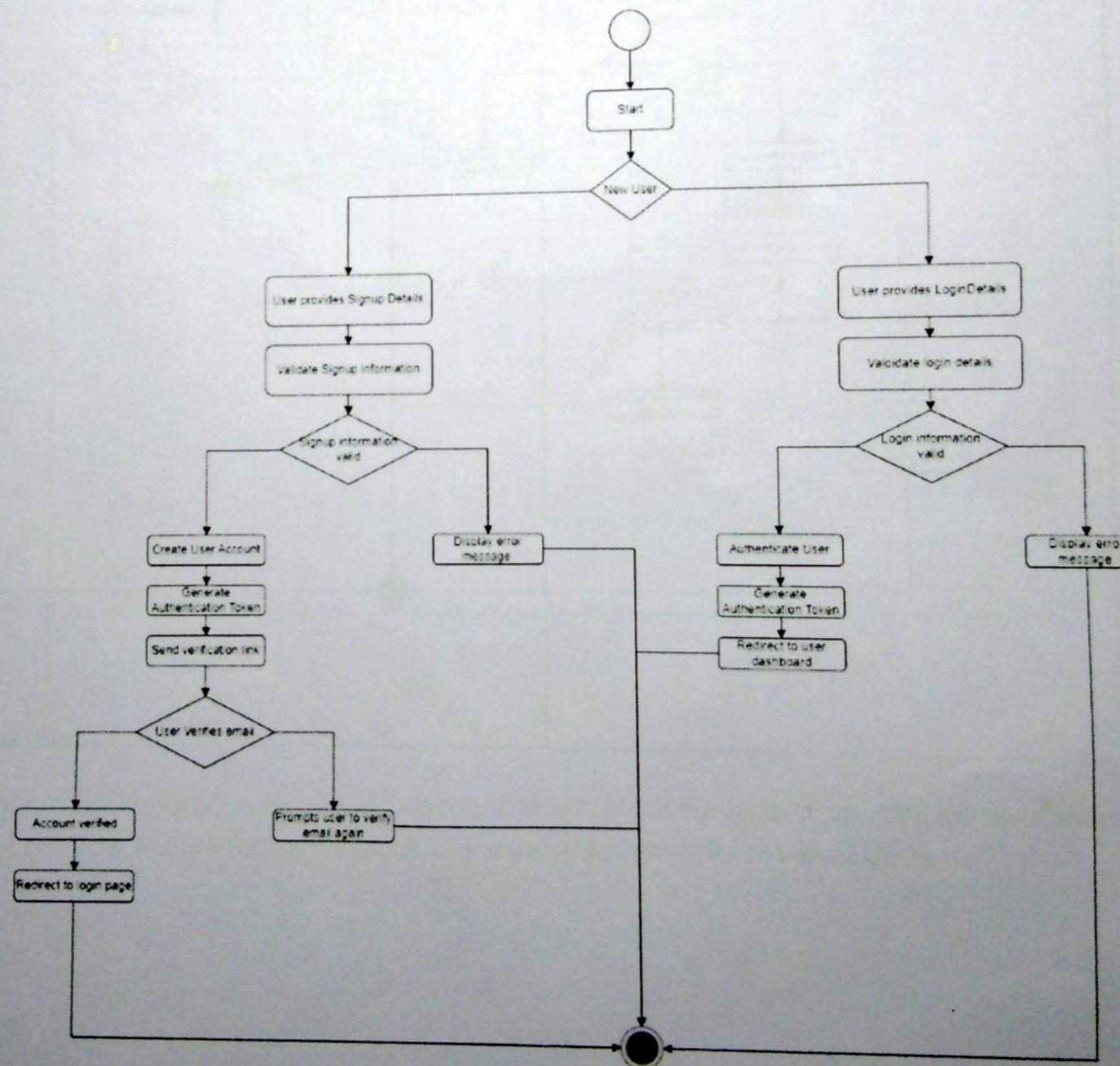
Amartya Mishra 60004210210

Gaurang Bhogle 60004210192

Shubham Mehta 60004210191

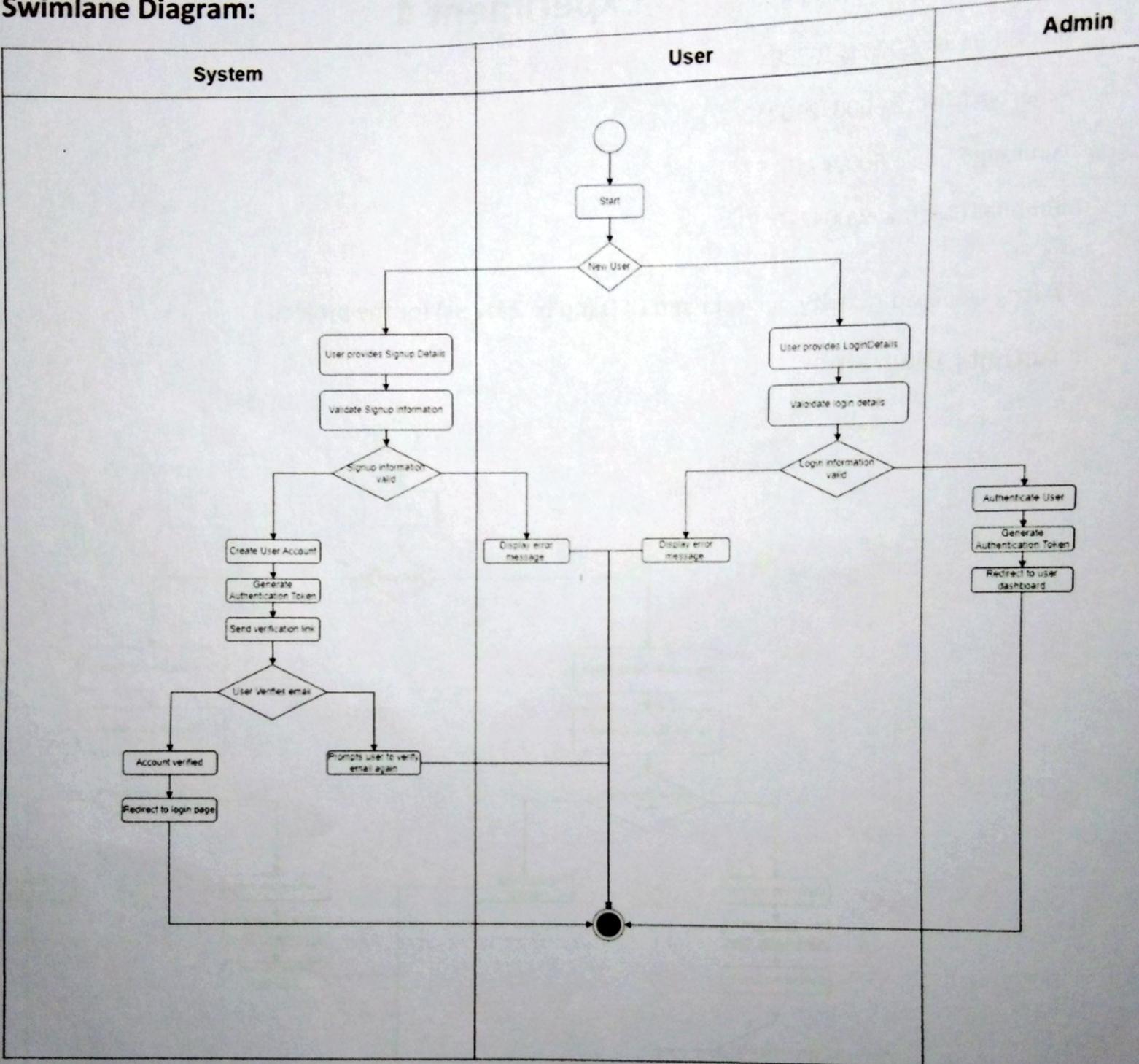
Aim: Develop Activity diagram and DFD (up to 2 levels) for the project.

Activity Diagram:

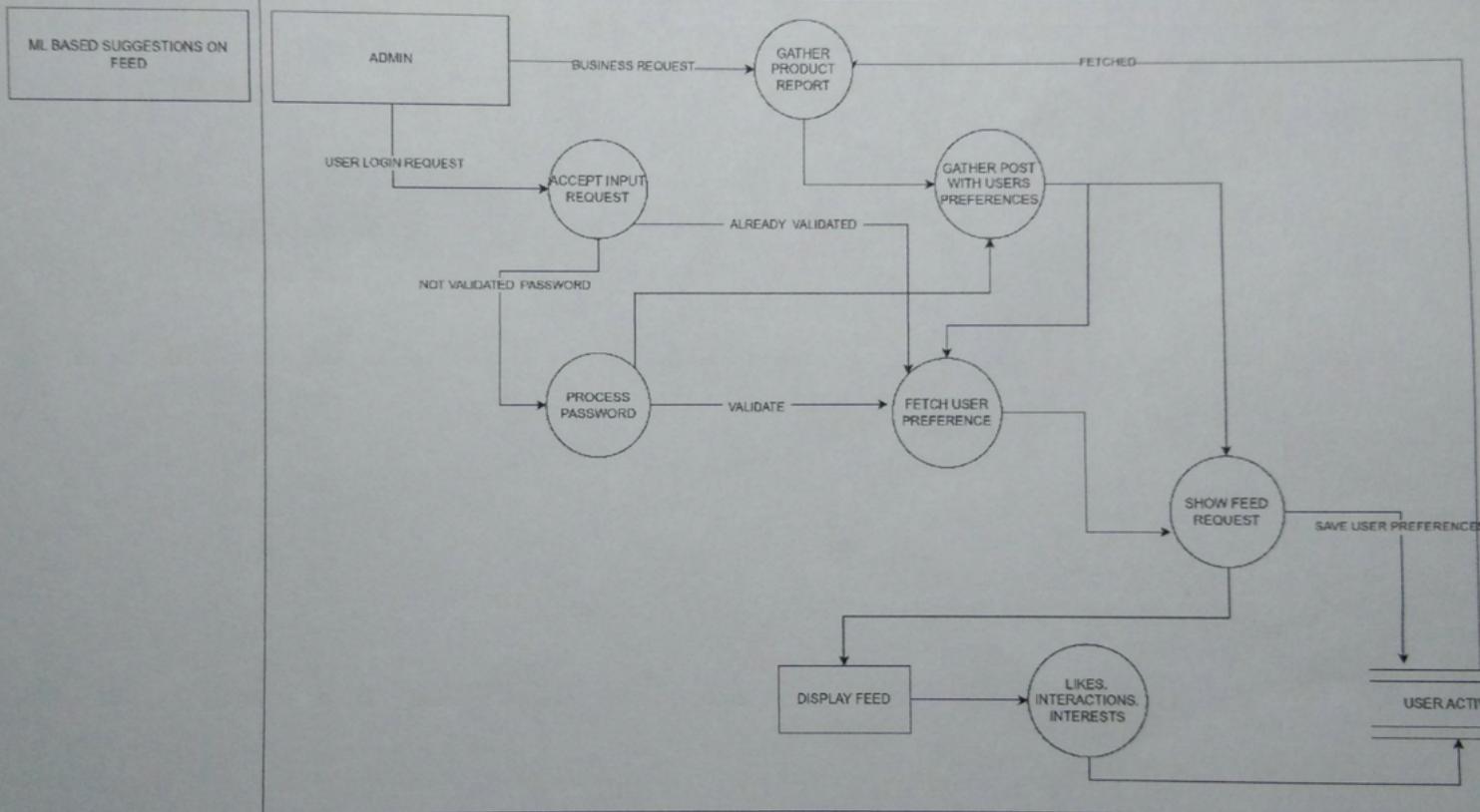
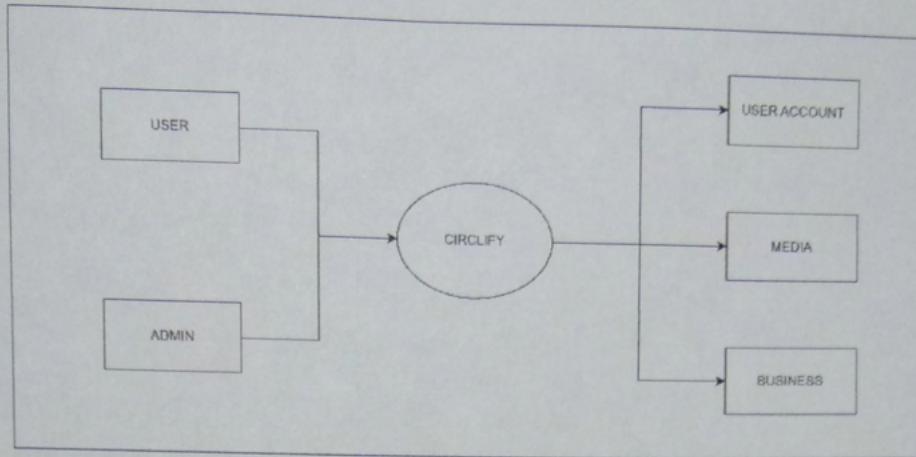


Swimlane Diagram:

Admin



DFD:



Conclusion:

Thus, we are able to draw Activity and Swim lane diagram for our case study. We are also able to depict the flow of data through various processes through different level DFDs

SE EXPERIMENT-5

Aim: Develop Sequence, Collaboration & state diagram for the project.

Theory:

Sequence diagram:

- screen listener: Allows user to interact with the app through physical touch inputs.
- feed: Display the recent post & content most interacted with.
- Post: content shared by user in form of ~~more~~ audio, video, ~~audio~~ etc. visible on app's feed for others to view.
- Media: Media in form of audio, video or both, can be shared by user using various methods.
- Profile: provides details of user's personal & public information. for logged in user this page also has options to edit the profile.

State Diagram:

- Signup login: It's the initial destination for new users where new creds are created.
- Scrolling: It becomes accessible post credential validation. allowing users to explore their feed.
- user can access feature of Direct message (DM) to share text & other media in form of post or normal uploads from device itself.

- Profile feature : Enables user to view & edit their own profile details. user can make security changes, password changes etc here as well.
- users can engage with media by sharing stories liking post & performing various preferences update within the platform to access content the person likes more frequently.

Conclusion : Thus we are able to draw sequence collaboration & state diagram.

Experiment 5

Darshit Sarda 60004210208

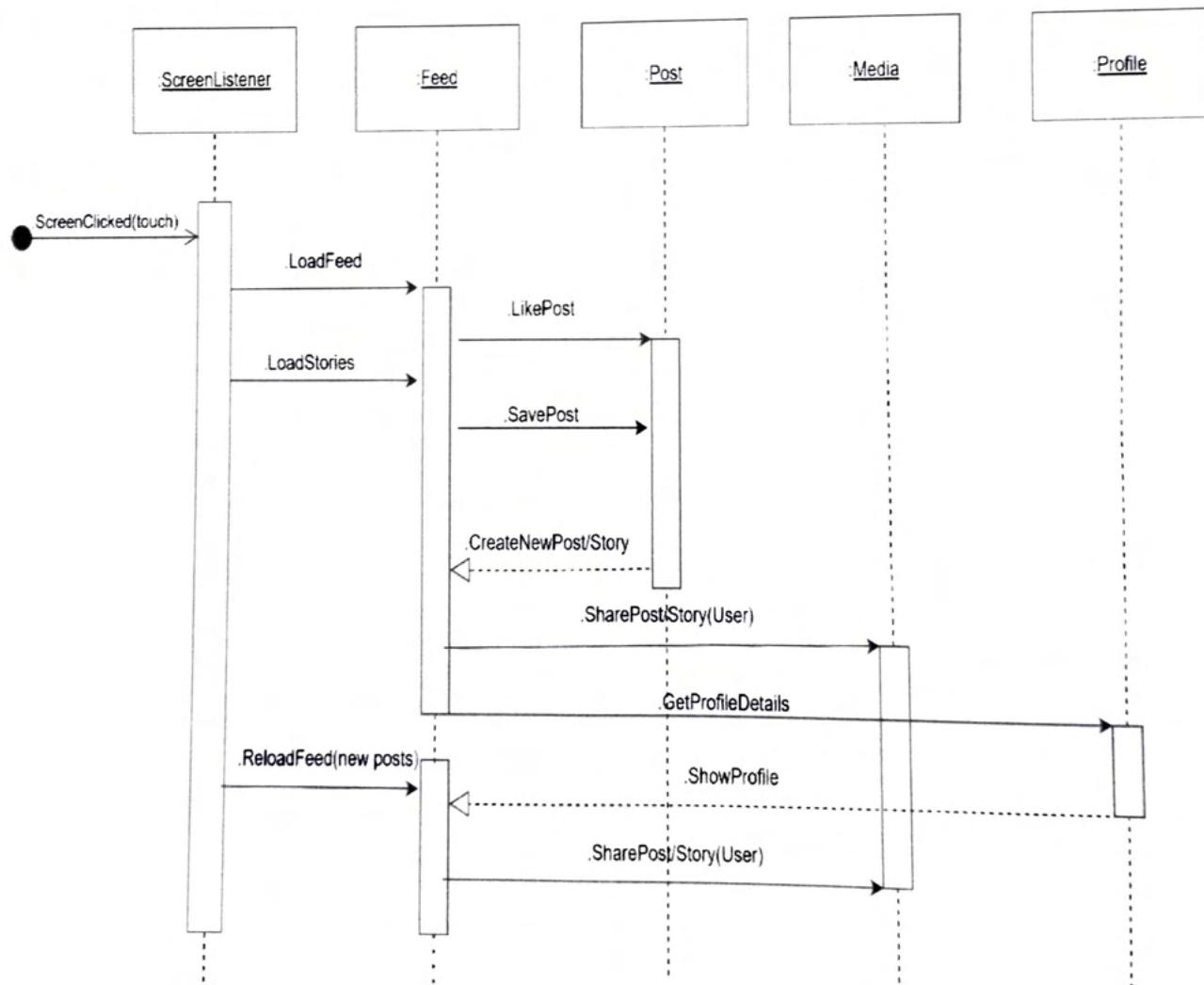
Amartya Mishra 60004210210

Gaurang Bhogle 60004210192

Shubham Mehta 60004210191

Aim: Develop Sequence, Collaboration and State diagram for the project.

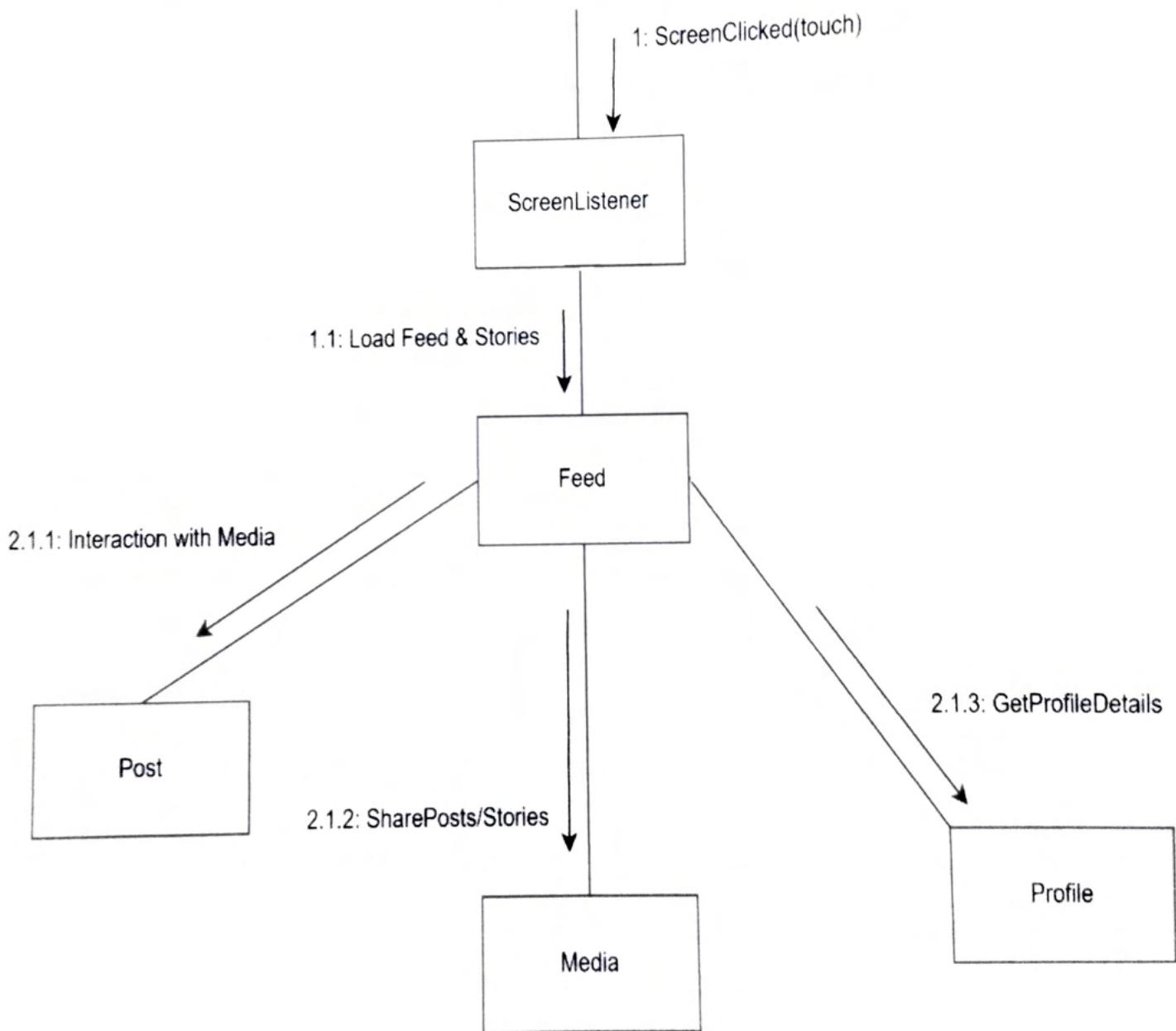
Sequence Diagram:



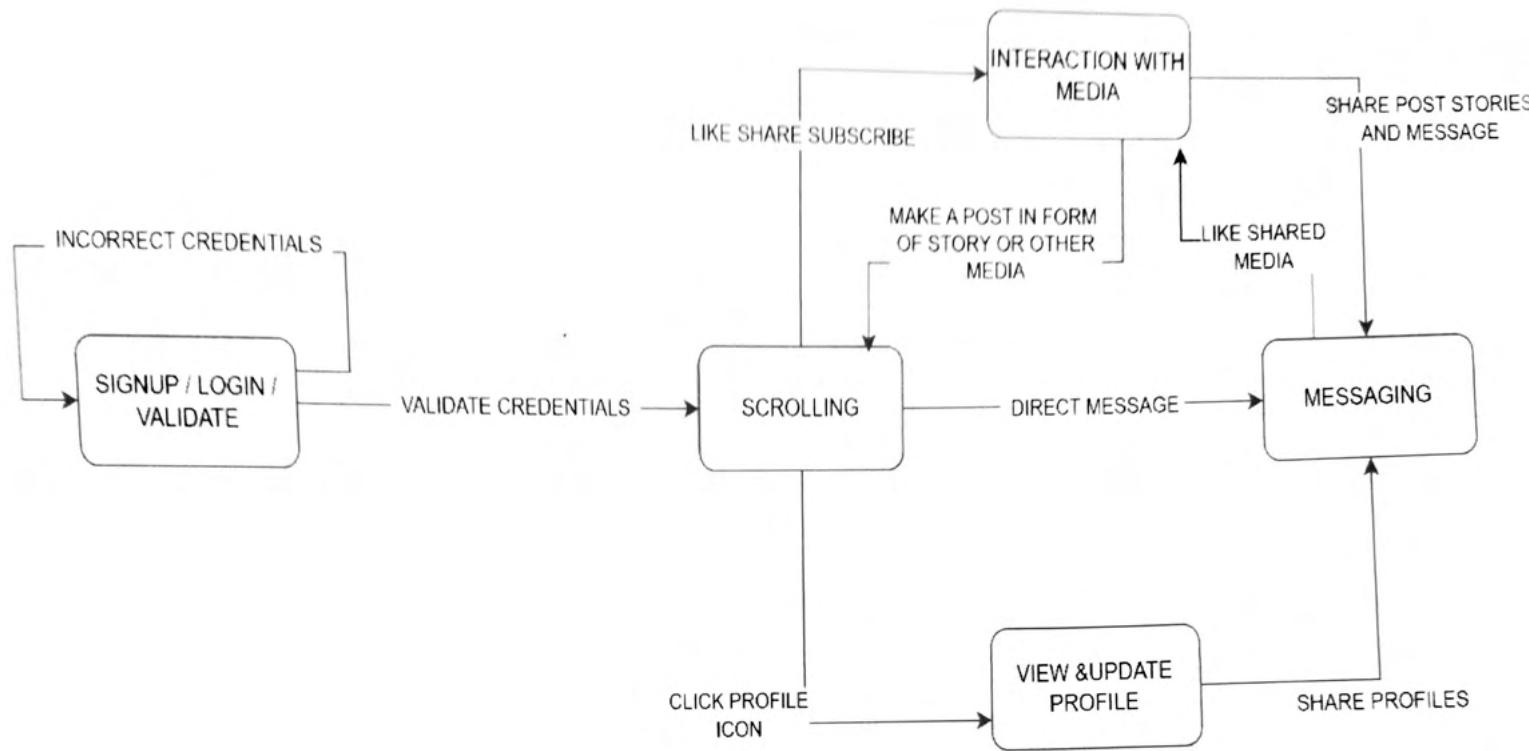
- ScreenListener: Touchscreen for interacting with the app by physical touch
- Feed: Loading the recent posts of the accounts you follow
- Post: Photos/videos posted by user that shows up on the feed

- Media: Sharing posts/stories (Media) to different users
- Profile: Details about every individual's profile on Circlify

Collaboration (Communication) Diagram:



State Diagram:



- Signup/login/validation page is the landing page for new users
- Scrolling can start after user has validated credentials and follows different accounts, consequently having posts load up on their screen
- User can share posts to other users via DM (direct message)
- User can click on profile to view and update their profiles
- User can also interact with media: sharing stories, liking posts etc.

Conclusion:

Thus, we are able to draw Sequence, Collaboration and State diagram for a functionality of our case study.

SE - Experiment 6

60004210210

Amartya MISHRA
COMPS - C31

Aim: Estimate effort & cost required for project using FPI COCOMO. Create WBS & Gantt chart for the same. Use PM tool to depict a plan.

Theory:

FP calculation:

Info Domains	Value	Cnt	weight factor		
			Simple	Avg	Complex
EI's		3	3	4	6
EO's		2	4	5	7
EQ's		1	3	4	6
ILF's		4	7	10	15
EIF's		1	5	7	10

count

3 → External Inputs (EI's) : User login request, business request , Accept output request .

2 → External outputs (EO's) : Show feed request, Save user preference .

1 → External Enquiries : Gather Post with user preference

4 → Internal logic files: User Account, Activity, Media, Business

1 → External Interface files: ML Based suggestions on feed.

Cost drivers:

- 1) Does system require Reliable Backup & recovery?
- 2) Are specialized Data communication required
- 3) Are there distributed processing functions?
- 4) Is performance critical?
- 5) Will system run in heavily utilized Environment?
- 6) Does the system require Online Data Entry?
- 7) Is Online Data Entry happening via Multiple screens & function?
- 8) Are IFI's updated online?
- 9) Are input output file inquiries complex?
- 10) Is the internal processing complex?
- 11) Is the code reusable?

$$\sum F_i = 59$$

$$FP_{\text{estimated}} = \text{count} - \text{total} \times [0.65 + 0.01 \times (\sum F_i)]$$

$$\text{count total} = 53$$

$$\begin{aligned} FP &= 53 \times [0.65 + 0.01 \times 59] \\ &= 66 \end{aligned}$$

$$\text{Proj Duration: } 3 \text{ months} = 22 \text{ person months}$$

$$\text{No. of Individuals} = 4 \therefore 22/4 = 5.5 \text{ PM/PI}$$

Conclusion: Thus we are able to calculate the efforts required for our project & prepare a Gantt chart.

Experiment 6

Darshit Sarda 60004210208

Amartya Mishra 60004210210

Gaurang Bhogle 60004210192

Shubham Mehta 60004210191

Aim: Estimate effort and cost required using FP/COCOMO for the project. Create WBS and Gantt Chart for the same. Use PM Tool to depict a project plan.

FP Calculation:

Information Domain Value	Count	Weighting Factor		
		Simple	Average	Complex
External Inputs (EIs)	3	3	4	6
External Outputs (EOs)	2	4	5	7
External Inquiries (EQs)	1	3	4	6
Internal Interface Files (ILFs)	4	7	10	15
External Interface Files (EIFs)	1	5	7	10

The counts for each component in the Function Point Analysis table were derived from the information provided in the data flow diagram.

External Inputs (EI):

Count = 3

- User Login Request
- Business Request
- Accept Input Request

External Outputs (EO):

Count = 2

- Show Feed Request
- Save User Preferences

External Inquiries (EQ):

Count = 1

- Gather Post with Users Preferences

Internal Logical Files (ILF):

Count = 4

- User Account
- Media
- Business
- User Activity

External Interface Files (EIF):

Count = 1

- ML Based Suggestions on Feed

The count represents the number of unique instances or occurrences of each component type identified from the data flow diagram. For example, there are three distinct External Inputs shown in the diagram, so the count for External Inputs is 3.

It's important to note that the accuracy of these counts depends on the level of detail and completeness of the information provided in the data flow diagram. As the project progresses and more detailed requirements are gathered, these counts may need to be adjusted accordingly.

Considering weighting factor as simple

$$FP \text{ estimated} = \text{count-total} \times [0.65 + 0.01 \times (\sum F_i)]$$

$$\text{count-total} = 3*3 + 2*4 + 1*3 + 4*7 + 1*5$$

$$= 53$$

14 questions:

1. Will the application use data communications?

- Yes (High Precededness) - 4

2. Are data or functions distributed?

- No (Low Required Reusability) - 2

3. Are there specific performance objectives that must be met?
 - Yes (High Architecture/Risk Resolution) - 5
4. Will the application run on a heavily used configuration requiring special design considerations?
 - Yes (High Architecture/Risk Resolution) - 5
5. Will the transaction rate of the application be high?
 - Yes (High Product Complexity) - 5
6. Will there be on-line data entry?
 - Yes (High Product Complexity) - 5
7. Will the application be designed for end-user efficiency?
 - Yes (High Product Complexity) - 5
8. Will there be on-line updates?
 - Yes (High Product Complexity) - 5
9. Is complex processing logic involved?
 - Yes (High Product Complexity) - 5
10. Is there an intent to provide usability for other applications?
 - Yes (High Product Complexity) - 5
11. How important are installation ease and conversion?
 - Moderate (No specific information provided) - 3
12. How important is operational ease?
 - High (No specific information provided) - 4
13. Will the application be accessed from multiple sites?
 - No (Single Site development) - 1
14. Is there an intent with the design to facilitate change?
 - Yes (High Development Flexibility) - 5

$$\sum F_i = 59$$

$$FP \text{ estimated} = 53 * [0.65 + 0.01 * 59]$$

$$= 66$$

Our project is having FP of 66

To calculate the productivity for each person based on the given information, we can follow these steps:

1. Calculate the total person-months required for the project using the given Function Points and project duration.
2. Divide the total person-months by the number of individuals working on the project to determine the average productivity per person.

Calculate total person-months:

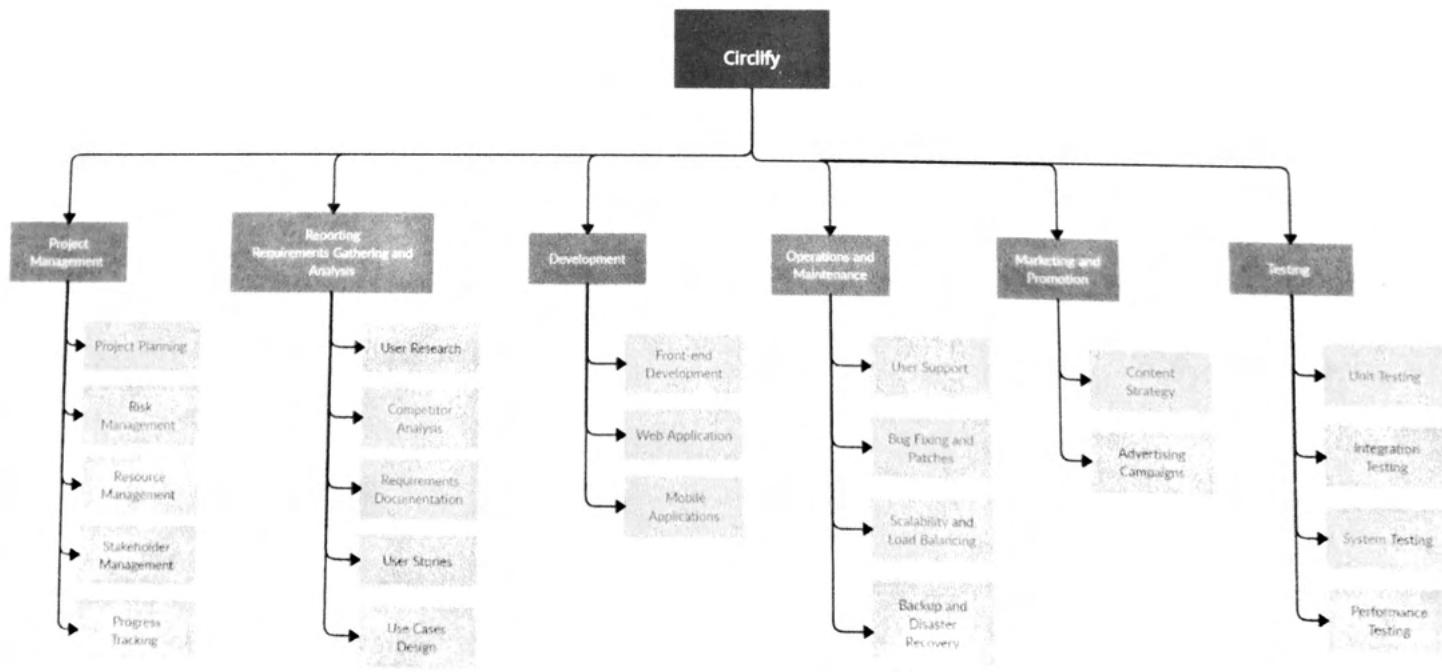
- Total Function Points (FP) = 66
- Project Duration = 3 months
- Total Person-Months = Total Function Points / Project Duration = $66 \text{ FP} / 3 \text{ months} \approx 22 \text{ person-months}$

Determine average productivity per person:

- Number of Individuals = 4
- Productivity per Person = Total Person-Months / Number of Individuals = $22 \text{ person-months} / 4 \approx 5.5 \text{ person-months per person}$

So, based on the given information, the average productivity per person for the project is approximately 5.5 person-months.

Work Breakdown Structure:



Gantt Chart:



Conclusion: Thus, we are able to estimate effort required for our project and also create Gantt Chart.

Circlify social media platform

Circlify SRS document

Amartha

4-Feb-14

10-Feb-14

Test scenario description	Importance	No. of test cases
login signup for existing users	High	3
post media	Medium	
share media in form of audio videos and text	Medium	3
send follow request and accept request	Medium	3
ad reports (test whether ads have caused any change in number of purchases of the product)	Low	3

	Circlify social media platform circlify SRS document
	Amar Tyaa
	1-Jan-24
	29-Mar-24

Test case ID	Test Objective	Precondition	Steps:	Test data	Expected result	Post-condition
TC_1.1	To verify if a user with valid credentials can successfully login.	1. A valid User account exist and is registered 2. Good internet connection	1. In the login Panel, enter the username 2. Enter the password	"A valid username" Enter the actual data in your real time situation	The user is logged in successfully. There is only one expected result for the entire test case.	For first time users personal information is displayed. Note: This info is only additional. Just a pointer to the tester
TC_1.2	password valid	1. Correct username inserted in the field	1. In the login Panel, enter the password	"A valid password format with specified length"	An Error message is displayed and the user is not logged in to the Orange HRM portal. "<exact error Message>"	As you can see, post condition can be left empty when there is nothing else add
TC_1.3	To verify if a user can sign up for a new account.	1. A User name to login to be available for a first time user of the site. 2. site is launched on a compatible browser/device	1. In the login Panel, enter the username 2. Enter the Password for the User account in the password field 3. Click "Signup" button	"A valid username" "A valid Password" The user is Registered successfully and the personal information page is displayed for further details		User account is created and the user registered in the Database
TC_2.1	Ability to upload media in form of image video and audio	1. User is logged in 2. Required permissions are granted to the application	1. Ask for permission to upload media from device 2. Select the media from the device menus 3. Click on "create post"	sample data to upload in audion video and image format	Permission granted thus access device data	
TC_2.2	Check the upload of image	Permissions granted to access device storage and media	select photo in a valid Ijpeg format Choose a image file of type "JPG" that is less than 1 MB Click on upload	Test images in valid format Name of the image Location path on the machine	The first name field needs to now show the new value entered The editing image field is visible	
					The file gets uploaded	

TC_2.3	Check the upload of video	1. User logged in 2. A valid video to upload that	select a valid video from device		The edit video options is displayed	
			Click on upload		The file gets uploaded and the older image is replaced	
TC_3.1	Test if user is able to send post	1. Valid user login 2. User1 sending the post follows User2 receiving the post	Click on send post	A sample post	Share media screen shows up with user names of account User1 can send post to	
TC_3.2	Test if user able to send stories	1. Valid user login 2. User1 sending the story follows User2 receiving the post	Click on share story	A sample story	Share media screen shows up with user names of account User1 can send post to	
TC_3.3	user able to direct message some one	1. Valid user login 2. User1 sending the post follows User2 receiving the message	Click on send Direct message icon	A sample account to receive and send text	The Direct messaging page shows up with all the account whom User1 has recently contacted	A messaging thred is created for User1 and User2
TC_4.1	Test if user is able to send follow request to User2	1. Valid user login 2. User1 sending the post follows User2 receiving the post 3. User2 has a account on the platform	1. Search for user account 2. On selecting the account click on send request	user account to send request to	the page updates the button of send request to sent	A request thread is created which shows User1 is sending request to User2
TC_4.2	Test If user able to accept follow request	1. Valid user login 2. User1 sending the post follows User2 receiving the request.	1. Click Accept request 2. Show option of Follow back if User2 not already following the User1.		The User1 is added as a follower to person accepting the result.	
TC_4.3	Test If user able to remove followers and unfollow	1. Valid user login 2. User1 sending the post follows User2 receiving the request. 3. User1 following User2 and viceversa.	1. Select User2 account 2. Show option of Unfollow User2. 3. Click on the unfollow button		Unfollow leads to deletion of relation of User1 User2 follower following instance	
TC_5.1	track user activity	1. User agrees to sharing information to app 2. User has some activity on the app	1. User Post and story interactions are recorded 2. User likes and dislikes are recorded 3. Report based on what user app usage patterns are generated.	Sample User activity data	User activity is recorded and report is generated	
TC_5.2	test if ads worked	1. User has interacted with ads 2. business individuals have provided the data of customers aquired through Circlify	1. User ad interactions are recorded 2. User likes and dislikes are recorded 3. User buying patterns pattern algorithms discovered by the algorithm	Sample User activity data	User activity is recorded and report is generated With detailed analysis of ads impacting user buying patterns	
TC_5.3	push ads based on activity	1. User has interacted with ads 2. business individuals have provided the data of customers aquired through Circlify	1. Algorithms filter the relevant ads from the various business stake holders 2. Based on filtered ads select ads which are best suited for the user .	sample ads and user activities matching the ad categories	Track the ads displayed and await the report for these ad performances to submit to business stake holders	

SE - Experiment - 8

60004210210

Amartya Mishra
COMPS - C31

Aim: To create a RMMM plan. Create Risk assessment template for a case study.

Theory:

Risks:

- Compatibility issues across various devices & OS.
- Poor quality Documentation.
- Under Estimating Data size of user Data & interaction
- absence of clear monetization strategy.
- lack of structured change control process
- Change in requirements
- Lack of Development Experience
- deviation from Software Standards.
- late delivery
- significant evolution of customer requirement while development

Risk Table:

Risk	Category	Prob	Impact
Compatibility issue	TI	75	1
Poor documentation	BU	75	1
under estimate Data size	PS	70	2
----- cutoff -----			
unclear Monetizat^n strategy	BU	45	2
change in Requirement	PS	35	2
lack of dev Exp	TI	20	3
late delivery	BU	20	4
change of Business Req.	CR	15	4

Risk for RMM: under estimating Data size for user Data & under action.

RE (Risk Exposure) : Prob of Risk x Impact of Risk

$$RE = 0.1 \times \$50,000$$

$$RE = \$35,000$$

Conclusion:

We successfully created A RMM Plan.

Risk Id : 10001010DBS	Date: 15/02/2024	Probability : 80%	Impact: High
Description: Underestimating the necessary database size could lead to potential performance issues, data loss, or system crashes due to insufficient storage capacity.			
Refinement/ Context:			
<ol style="list-style-type: none"> 1. Subcondition1: Initial user base and the reach of the app was underestimated with incorrect estimation of user base size. 2. Subcondition2: The design of system was calibrated to cater to 100000 request per second which might be less than the current request rate. 3. Subcondition3: The amount of media uploaded is unrestricted leading to overflowing memory and loss of uploaded media and affecting the application features. 			
Mitigation / Monitoring:			
<ol style="list-style-type: none"> 1. A team of few members to actively monitor the growth rate of the database. 2. Develop an algorithm to predict the time before capacity overflow of database 3. Implement data limits on users to limit the amount of data being uploaded by the user 			
Management/ Contingency plan / trigger:			
<ol style="list-style-type: none"> 1. The team cost approximates INR 1, 20, 000 given a team of size 3 and hiring new developers who can develop the algorithm to monitor and predict the data overflow date. 2. For maintaining Cloud services to altogether avoid the given situation the total cost incurred will be INR 4, 00, 000. 			
Current Status:			
Mitigation steps initiated.			
Originator: Darshit , Gaurang		Assigned: Amartya , Shubham	



SE Expt. 9

Darshit Sarda -60004210208

Amartya Mishra-60004210210

Gaurang Bhogle-60004210192

Shubham Mehta-60004210191

Aim: Study of Configuration Management using GitHub.

Theory:

- ❖ To demonstrate and resolve a conflict in a GitHub repository with diagrams and an SRS uploaded, you can follow these steps:
- ❖ Create a Conflict: Make changes to a file in your repository from two different branches. For example, you can edit the SRS file in one branch and update a diagram in another branch.
- ❖ Commit Changes: Commit the changes to each branch separately. Ensure that the changes conflict with each other, meaning they modify the same lines of code or content in the files.
- ❖ Merge or Pull Request: Merge the branches or create a pull request to merge one branch into another. This action will trigger a conflict if changes from both branches affect the same lines in a file.

Resolve Conflict:

- Open the affected file(s) in your preferred text editor or integrated development environment (IDE).
 - Locate the conflicting lines marked by Git. These lines will typically have both versions of the changes from each branch, surrounded by conflict markers (<<<<<, =====, >>>>>).
 - Edit the conflicting lines to resolve the conflict, keeping the changes that you want to keep or combining them as needed.
 - Remove the conflict markers (<<<<<, =====, >>>>>) once you have resolved the conflict.
 - Save the changes to the file(s).
- ❖ Commit Changes: After resolving the conflict, commit the changes to complete the merge or pull request process.
 - ❖ Push Changes: Push the merged changes to your GitHub repository to update it with the resolved conflict.
 - ❖ Verify: Verify that the conflict has been resolved correctly by reviewing the files and ensuring that they contain the intended changes from both branches.

COCOMOMO Private

main 1 Branch 0 Tags

shubhammehta39 Add files via upload

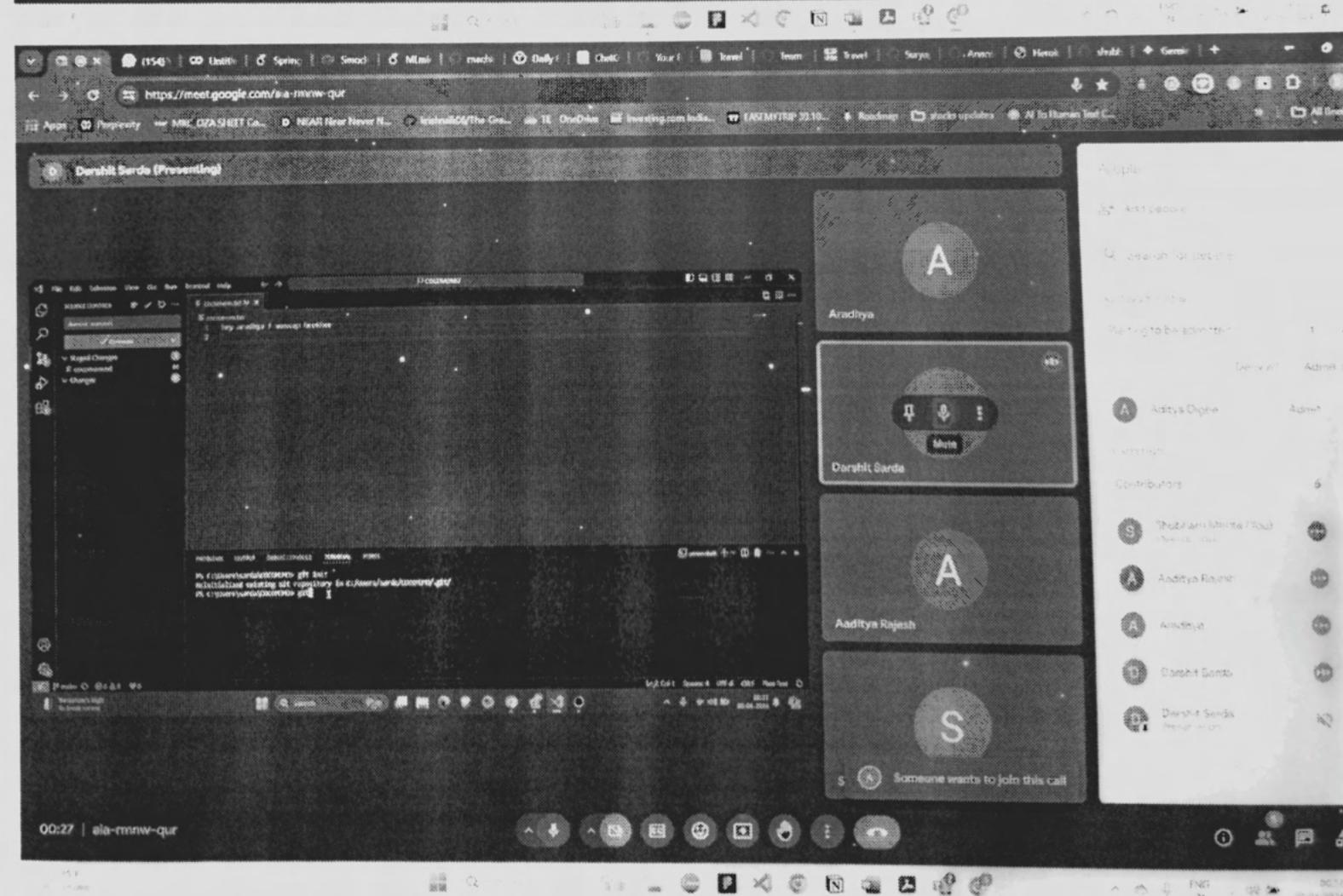
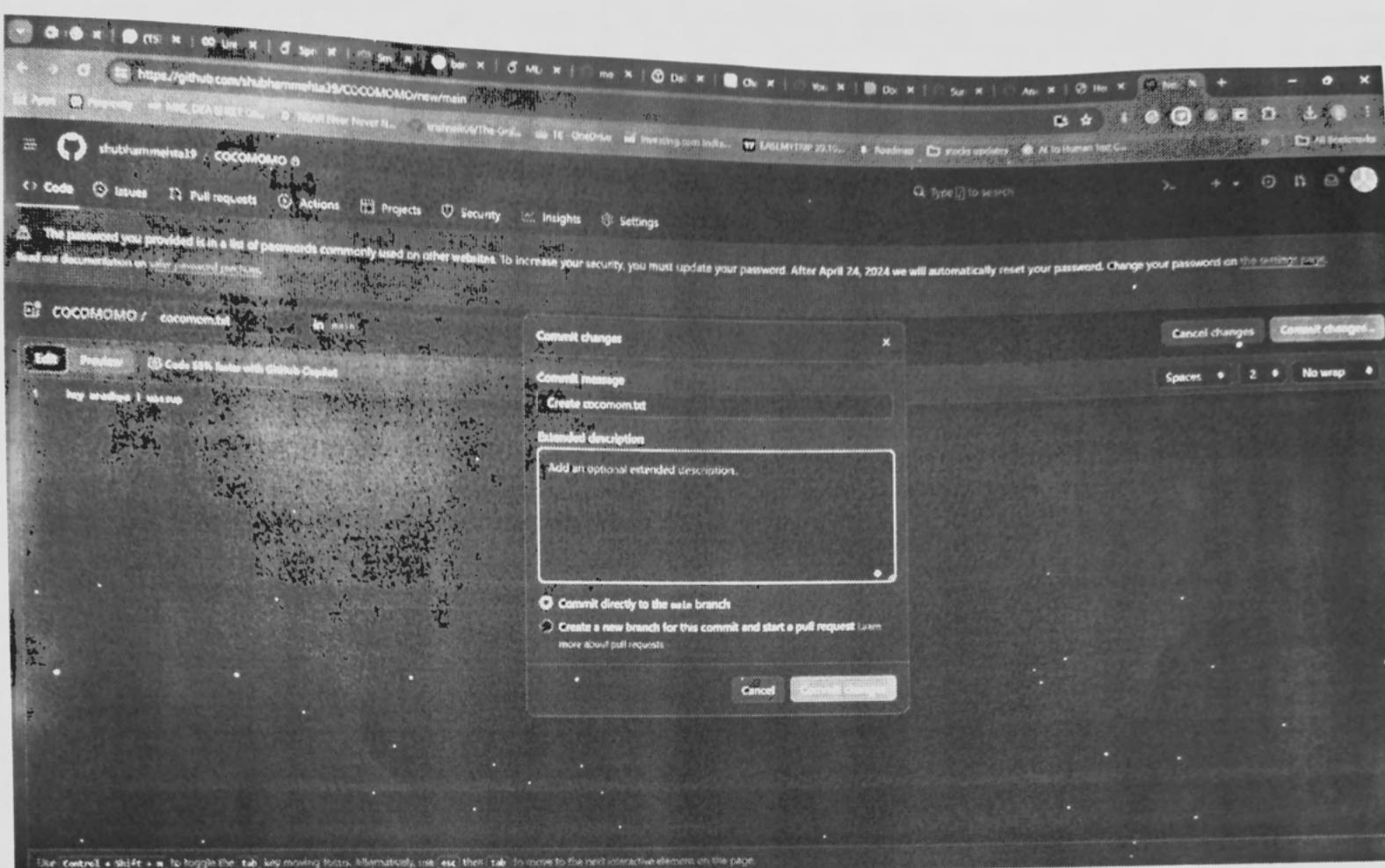
File	Commit Message	Time Ago
README.md	Initial commit	9 minutes ago
SE_Exp09_Circlify.pdf	Add files via upload	7 minutes ago
SE_Exp10_Circlify.pdf	Add files via upload	7 minutes ago
SE_Exp1_Circlify.pdf	Add files via upload	7 minutes ago
SE_Exp2 SRS_Circlify.pdf	Add files via upload	7 minutes ago
SE_Exp3_Circlify.pdf	Add files via upload	7 minutes ago
SE_Exp4_Circlify.pdf	Add files via upload	7 minutes ago
SE_Exp5_Circlify.pdf	Add files via upload	7 minutes ago
SE_Exp6_Circlify.pdf	Add files via upload	7 minutes ago
SE_Exp7_SampleTest-Cases-Circlify.xlsx	Add files via upload	7 minutes ago
SE_Exp7_SampleTest-scenariosCirclify.xlsx	Add files via upload	7 minutes ago
SE_Exp8_Circlify.pdf	Add files via upload	7 minutes ago

Read our documentation on [safer password practices](#).

cocomomo / cocomom.txt in main

Edit Preview Code 55% faster with GitHub Copilot

1 hey aradhya I missup



Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks or learn more about diff comparisons.

base: main ⌛ compare: digbu ✅ Able to merge. These branches can be automatically merged

Create pull request

Discuss and review the changes in this comparison with others. [Learn about pull requests](#)

1 commit 1 file changed 1 contributor

Commits on Apr 30, 2024

Create cocomomm.txt

shubhammehta39 committed now

Verified 2007146 ⌂

Showing 1 changed file with 1 addition and 0 deletions.

Split Unified

1 README.md cocomomm.txt

(0) -b 0 +1 00
+ hey digbu boy.

https://github.com/shubhammehta39/COCOMOMO/compare/main...digbu

The password you provided is in a list of passwords commonly used on other websites. To increase your security, you must update your password. After April 30, 2024 we will automatically reset your password. [Change your password](#) on the settings page.

Read our documentation on safer password practices.

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks or learn more about diff comparisons here.

base: main ⌛ compare: digbu ✅ Able to merge. These branches can be automatically merged

Add a title

Create cocomomm.txt

Add a description

Write Preview

Add your description here...

Markdown is supported ⌂ Paste, drop, or click to add files

Reviewers None yet

Assignees No one assigned yet

Labels None yet

Projects None yet

Milestones No milestones

Development Use [Closing keywords](#) in the description to automatically close issues

Helpful resources GitHub Community Guidelines

1 commit 1 file changed 1 contributor

Create cocomomm.txt #1

shubhammehta39 wants to merge 1 commit into main from digits

Conversation Commits Checks Files changed

shubhammehta39 commented now
No description provided.

Update cocomomm.txt Verified 200718

Add more commits by pushing to the digits branch on shubhammehta39/COCOMOMO.

Require approval from specific reviewers before merging Pulses ensure specific people approve pull requests before they're merged. Add note

This branch has no conflicts with the base branch Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command line instructions.

Add a comment Write Preview

Add your comment here... Markdown is supported Paste, drop, or click to add files

Reviews No reviews. Still in progress! Learn about draft PRs.

Assigned No one — assign yourself!

Labels None yet

Projects None yet

Milestones No milestones

Development Successfully merging this pull request may close these issues.

Note yet

Notifications 1 participant

Customize

Unsubscribe You're receiving notifications because you're watching this repository.

cocomom.txt

Incoming 7ce2a83 - origin/main, origin/HEAD

Accept Incoming | Accept Combination | Ignore

1 hey aradhyaa sakalley I wassup

2

Current 73ae53d - main

Accept Current | Accept Combination | Ignore

1 hey aradhyaa I wassup brottie *

2

Result cocomom.txt

No Changes Accepted

1 hey aradhyaa I wassup

2

1 Conflict Remaining

Complete Merge

Conclusion

Hence we successfully studied and implemented configuration management using GitHub.

Aim. Study of DevOps

Theory:

- A no. of online Devop tools are available which are Open ~~source~~ source & free to use
 - select set of tools for the various stages of SW dev makes a ~~&~~ Devops tool chain
 - Diff. tools are used in different stages.
- Collaboration: when a team working across different part of project is able to create & manage the project at once, it's done using collaborative tools
The Tool used is Docker / skype / slack
- Planning: need to plan the development of product across various phases (ASANA, clarizen)
- Source control: The base version is fixed, changes are made on new Branches & merged (GIT / SVN Version)
- Issue tracking: Tracking issues & Point of failure possible using DevOps (Jira, Zendesk)
- config management: auto config tools avail (Ansible / Puppet / Chef)
- Continuous Integration - (Jenkins / team city)
- Binary Repositories (RT factory / Nexus / maven)
- Monitoring (Big Panda & Sensu)

- Automating all development & testing - Ansible (Terraform)
- Database integration. (Razor SQL, Team Desk)
- Development (Docker & IBM)

Conclusion : Thus we listed all the phases & tools used in the Devops tool chain.

SE - Assignment 1

60004210210

Amaranya Mishra
COMPS - C31

Q1 Elaborate Task set for creating component level design in OO project.

→ The Task set for creating ^{component} OO level design in OO projects

- 1) Identify all design corresponding to problem Domain
- 2) Identify all design classes that correspond to Infrastr. Dom.
- 3) Elaborate all design classes that are not acquired as reusable component.
 - (i) Specify message details when classes / components collaborate
 - (ii) Identify appropriate interfaces for each component
 - (iii) Elaborate attributes & define data types & Data-structure required to implement them.
 - (iv) Describe processing flow within each operation in detail.

4) Describe present DS & identify the classes required to manage them.

5) Develop & Elaborate Behavioral Representation for a class or component.

6) Elaborate Development diagrams to provide additional implementation Details.

7) factor every component level design Representation & always consider alternatives.

2) The golden Rule of user Interface are:

- i) visibility
- ii) Feed Back Feed Back
- iii) consistency
- iv) Flexibility.
- v) Simplicity

Methodology

- a) Place user in control
- b) Reduce users Memory load
- c) Make interface consistent

Define a Interaction modes in a way that does not force a user into unnecessary action.

Provide for flexibility in action

Streamline Interaction with increasing skill level & allow customized interaction.

Hide technical internals from casual users

Design for direct interaction with objects that appear on screen.

- d) Reduce users Memory load

Reduce demand on short term memory

Establish meaningful defaults.

Define short-cut Intuitives

Disclose information in progressive fashion

- e) Make interface consistent

Allow user to put current task into a meaningful context.

- (3) Transform mapping is a process used in Software Engineering to map data flow from source to destination.
- It is a set of design steps that allow a DFD with transform flow characteristics to be mapped onto a specified architecture style.
- To map DFD onto a software Architecture, we would undertake the following steps:
- 1) Review the fundamental system model.
 - 2) Review & refine Data flow diagram for software
 - 3) Determine whether DFD has transform transaction flow characteristics.
 - 4) Isolate the transform center by specifying incoming & outgoing flow boundaries.
 - 5) Perform first level factoring.
 - 6) Perform second level factoring.
 - 7) Refine first iteration Architecture using design heuristic for improved software
- (4) Transaction mapping involves mapping the flow of transaction within a system.
- It helps ensure that all necessary steps are identified & executed correctly.
- Step:
- (1) Identify logical Transactions
 - (2) Each logical transaction is mapped to a series of DB operation
 - (3) Transaction mapping ensures automatic Transaction

14) It involves optimizing performance of transaction

SE - Assignment 2

60004210 210

Amartya mishra
COMPS- C31

- (1) Key Devops fundamentals Revolve around the concept of continuous integration , delivery , automation & collaboration .
- Devops is more practice than technology thus many tools are applicable to different stages of development.
- A Number of open source Devops tools are present clubbing one of owned makes a dev-ops tool chain Thus product delivery more efficient .
- Different stages of sw dev:
 - 1) Collaboration
 - 2) Planning
 - 3) source control
 - 4) issue tracking
 - 5) configuration management
 - 6) continuous integration
 - 7) Binary Repository
 - 8) Monitoring
 - 9) Automated testing
 - 10) Development
 - 11) Database .
- Companies require skilled individuals familiar with such Dev Ops tools .
- Faster Development : since , using these tools most of process is automated

(2) Docker Architecture:

- Docker uses client server Architecture
- Docker client talks with Docker Daemon, Daemon Builds, Runs & distributes Docker containers
- Docker client & Daemon connect via REST API's over UNIX socket or network interface.

1) Docker client:

When user runs any command on Docker client CLI a REST Request forwards it to the daemon

The following commands can be run on CLI:

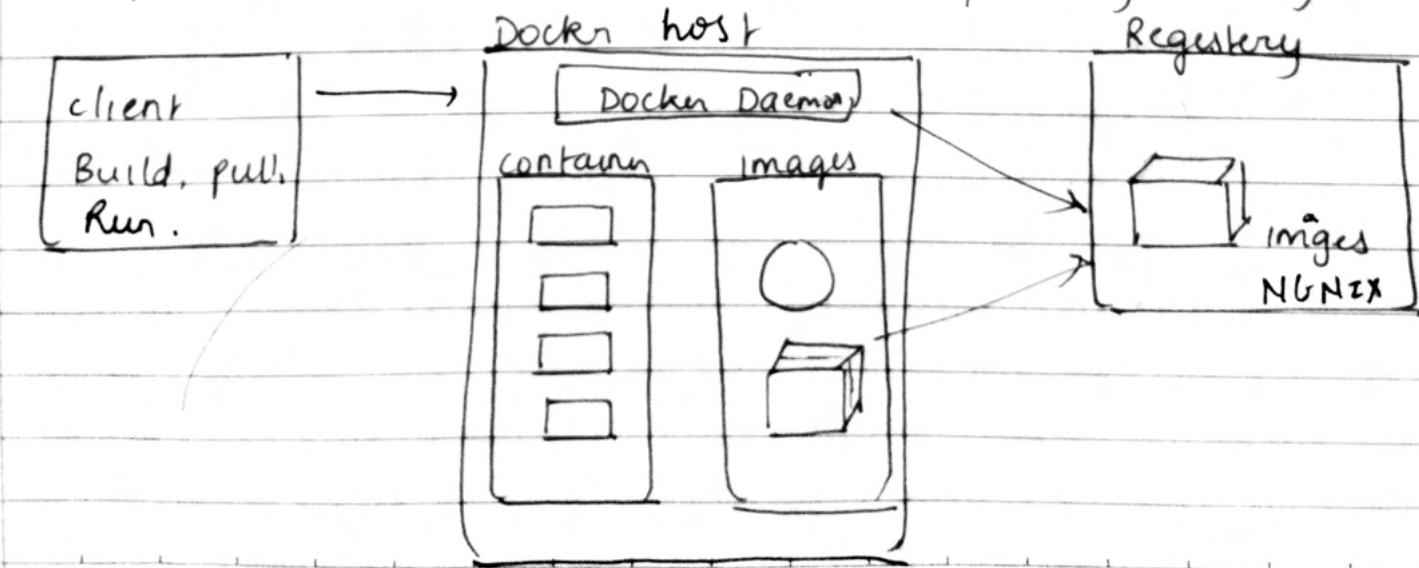
Docker Build

Docker pull

Docker Run

2) Docker host: It provides an environment to run the applications, It contains docker daemon, images, containers, Networks & storage

3) Docker Registry: It contains images. Images may be public (visible to all) or private (for only one org acm)



(3) Containerization : is a method of packing , running & distributing applications in containerized Environment / contained Environment . called a container . It allows isolation of application & its dependencies . Ensuring consistency & portability across different computing environments .

- Docker is a popular platform for containerization
- It provides tools & API's for creating deploying & managing containers . Docker containers package an application & its dependencies into standardized unit making it easy to deploy & run applications on any system that supports docker
- Docker uses layered file system where each layer represents the change to file system .
- Docker also provides Dockerfiles making it easy to manage container

Software Requirements Specification

for

Circlify

Version 1.0

Prepared by

Group Name: <place your group name here>

Amartya Mishra	60004210210	Amartyamishra2503@gmail.com
Gaurang Bhogle	60004210192	bgaurangan@gmail.com
Darshit Sarda	60004210208	darshitsarda10@gmail.com
Shubham Mehta	60004210191	shubhammehta2003@gmail.com

Instructor: Dr. Kiran Bhowmick

Course: Software Engineering

Lab Section: Software Engineering Lab

Teaching Assistant: NA

Date: 07-02-2024

Contents

REVISIONS	II
1 INTRODUCTION	3
1.1 DOCUMENT PURPOSE	3
1.2 PRODUCT SCOPE	3
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW.....	3
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....	4
1.5 DOCUMENT CONVENTIONS	4
1.6 REFERENCES AND ACKNOWLEDGMENTS.....	4
2 OVERALL DESCRIPTION	5
2.1 PRODUCT PERSPECTIVE	5
2.2 PRODUCT FUNCTIONALITY	6
2.3 USERS AND CHARACTERISTICS	6
2.4 OPERATING ENVIRONMENT.....	6
2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS	7
2.6 USER DOCUMENTATION	7
2.7 ASSUMPTIONS AND DEPENDENCIES	7
3 SPECIFIC REQUIREMENTS	8
3.1 EXTERNAL INTERFACE REQUIREMENTS	8
3.2 FUNCTIONAL REQUIREMENTS	10
3.3 BEHAVIOUR REQUIREMENTS.....	11
4 OTHER NON-FUNCTIONAL REQUIREMENTS	12
4.1 PERFORMANCE REQUIREMENTS.....	12
4.2 SAFETY AND SECURITY REQUIREMENTS.....	12
4.3 SOFTWARE QUALITY ATTRIBUTES	12
5 OTHER REQUIREMENTS	ERROR! BOOKMARK NOT DEFINED.
APPENDIX A – DATA DICTIONARY	ERROR! BOOKMARK NOT DEFINED.
APPENDIX B - GROUP LOG	14

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00



Academic Year: 2023_24

1 Introduction

1.1 Document Purpose

The Software Requirements Specification (SRS) document for Circlify serves as a foundational guide outlining the essential components and expectations for the development of the Circlify software product. It begins with an introduction providing context on Circlify's purpose and scope, followed by a delineation of its boundaries and included features within the scope section. Functional requirements are detailed, specifying the necessary capabilities and user interactions, while non-functional requirements outline quality attributes and constraints.

User interfaces and system architecture are described to illustrate how users will interact with the system and its underlying structure. Data requirements, assumptions, dependencies, and constraints are also documented, providing insights into Circlify's design considerations. Finally, acceptance criteria are defined to ensure stakeholder expectations are met. Overall, the SRS serves as a blueprint for development, facilitating communication between stakeholders and guiding the implementation process to ensure Circlify aligns with user needs and quality standards.

1.2 Product Scope

The Software Requirements Specification (SRS) for Circlify encapsulates the core features and expectations of a social media application designed for both individual and business users. Circlify enables seamless sharing of multimedia content such as audio, video, images, and files, fostering collaboration through posts, stories, and vertical videos. For businesses, Circlify offers a platform to promote ads and other targeted content to engage with their desired audiences effectively. The SRS outlines the application's scope, functionalities, user interfaces, system architecture, data requirements, and acceptance criteria. It serves as a blueprint for development, ensuring Circlify meets user needs and quality standards while facilitating communication between stakeholders and the development team.

1.3 Intended Audience and Document Overview

The Software Requirements Specification (SRS) document for Circlify is primarily directed towards the Circlify engineering and product team. Its secondary purpose is to align leadership and investors with the project vision and requirements. The detailed information provided within the SRS aims to provide the development team with a clear understanding of the expected functionality and characteristics of Circlify. This document serves as a comprehensive reference point throughout the development process, ensuring that all stakeholders are aligned and that the final product meets the defined requirements and expectations.

Following this introductory overview, Section 2 provides a high-level description of Circlify, its context, and primary components. Section 3 delves into the specific functional and behavioral requirements. Section 4 covers non-functional requirements around performance, security, and quality attributes. Finally, supporting appendices provide a data dictionary and project logs.



Academic Year: 2023_24

1.4 Definitions, Acronyms and Abbreviations

- API - Application Programming Interface
- SDK - Software Development Kit
- MVP - Minimum Viable Product
- SRS - Software Requirements Specification
- UI - User Interface

1.5 Document Conventions

In general this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1" margins found in this template. For Section and Subsection titles please follow the template.

1.6 References and Acknowledgments

Our app Circlify takes inspiration from various different applications in the market like Instagram, Facebook, Whatsapp, etc.

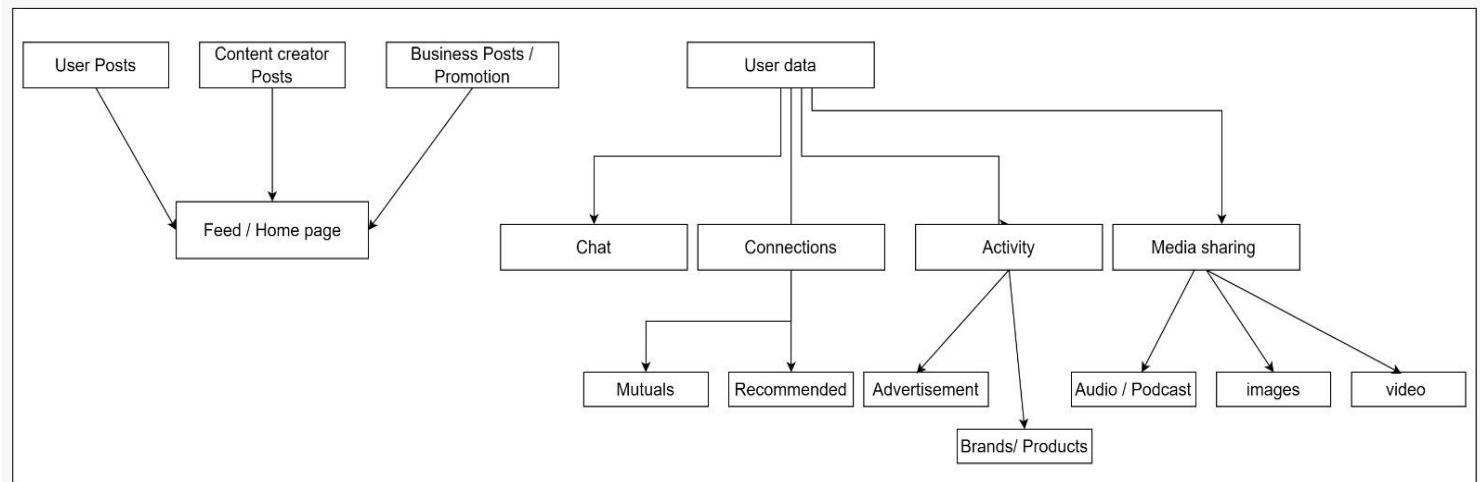
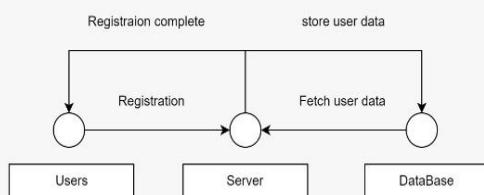


Academic Year: 2023_24

2 Overall Description

2.1 Product Perspective

Circlify will be designed as a standalone social application, not requiring integration with existing platforms. Users will access the platform via a web-based client and native mobile apps on iOS and Android. The backend will consist of cloud-hosted servers providing API-based access to a relational database.





Academic Year: 2023_24

2.2 Product Functionality

- User profiles, feeds, and social connections - Allows users to create personal profiles, connect with others, and view content feeds
- Sharing of posts with rich text, images, videos, and files - Enables users to share diverse post content including text, visual media, videos, and documents
- Ephemeral stories and vertical video sharing - Offers Snapchat-like ephemeral stories and TikTok-style vertical videos
- Multimedia upload and optimization - Provides uploading and automatic optimization of images, videos and other media
- Hashtagging and social discovery - Implements hashtags and recommendations for discovering content and like-minded users
- Business tools for promotions and analytics - Gives businesses abilities to run targeted promotions and view engagement analytics
- Configurable privacy controls and content moderation - Offers users granular controls over content visibility and moderates inappropriate content

2.3 Users and Characteristics

- Individual users - Share personal experiences and connect with friends
- Business users - Share promotions, ads, reach target audiences; carry out business research on set target audiences.
- Age groups 13 and above – comprising of teenage and young adults majoritarily.
- Technically apt and novice users

Primary users are individuals looking to share personal content and experiences.

2.4 Operating Environment

The Circlify application is being developed as a cross-platform mobile app for Android and iOS devices. It will target a minimum SDK version of 21 for Android and iOS 13 for Apple devices to maximize accessibility across a range of hardware. Considering the goal of broad user adoption, the app is designed to function smoothly on devices with as low as 2GB of RAM. Along with native mobile apps, users can also access Circlify via any modern web browser for platform flexibility.

On the backend, Circlify will employ a hybrid data persistence architecture. A relational SQL database will efficiently handle structured user profile information and metadata. A NoSQL document store will provide the flexibility to capture dynamic social content like posts and media. Security is paramount for a social platform, so rigorous policies will be enforced for access control, data encryption, and user authentication using industry best practices like OAuth and multi-factor mechanisms. The backend will provide a robust API layer to securely



Academic Year: 2023_24

expose functionality to clients. The overall technical approach aims to deliver an accessible, secure, and reliable experience across use cases.

2.5 Design and Implementation Constraints

Given the time and resource constraints of the initial development timeline, the product launch will strategically focus on implementing the core functionality and most critical user needs first. The engineering team will need to make careful design trade-offs and prioritize the MVP feature set guided by technical limitations. For example, while an advanced AI recommendation system is desired, it may be deprioritized for the first release in favor of essential capabilities like content sharing and user profiles. The cloud infrastructure will be scaled to efficiently support the target initial user base. Advanced monitoring and auto-scaling capabilities can be added later to manage growth. Where possible, the team will leverage existing open source libraries and reusable components to accelerate development. With agile methodology, the launch feature set will provide the essential user value while allowing for continuous enhancement and expansion in subsequent releases as future scope.

The other primary limitations are related to security of users and business personals . There will be a need of developing an adaptable security system that can adopt to age restricted content being circulated via the platform. There will bee need to verify the businesses that can actually promote and carry out business on the platform.

2.6 User Documentation:

The application's user documentation strategy encompasses various support channels, including in-app tooltips, FAQs, a help center, social community support, and a knowledge base on the website. In-app tooltips offer real-time guidance, FAQs address common queries, while the help center provides comprehensive resources. Social community support fosters engagement, and the knowledge base hosts detailed articles and guides. This multifaceted approach ensures users have diverse resources at their disposal, facilitating a seamless and productive user experience.

2.7 Assumptions and Dependencies

Key assumptions include:

- Projected user growth targets are accurate
- Core features can be built within budget and timeline
- Mobile app stores and ad networks will approve the application
- Target devices and browsers retain significant market share
- Users will be willing to adopt a new platform if it provides unique value
- Target demographics will respond positively to differentiating features



3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface shall consist of three different GUI screens. They are Home Page, Explore Tab and Chat Tab.

3.1.1.1 Home Page

- Conversation List and Chat Interface:
 - Users access ongoing conversations displayed with avatars or profile pictures and usernames.
 - Tapping opens a threaded chat interface for message viewing and replies.
- Messaging Features:
 - Text and media sharing, including photos and videos.
 - Emojis, stickers, voice messages, reactions, and likes enhance communication.
- Notifications and Privacy:
 - Notifications keep users updated on new messages.
 - Users control messaging privacy settings, managing who can message them and see their online status.

3.1.1.2 Explore/Search Tab

Personalized Content: Shows posts, stories, and updates from followed accounts.

Scrollable Interface: Allows vertical scrolling for continuous content viewing.

Engagement Features: Enables liking, commenting, and sharing directly from the feed.

Algorithmic Sorting: Prioritizes content based on user interests and engagement.

Search Page:

Content Discovery: Helps users find new accounts, posts, and relevant content.

Search Bar: Allows input of keywords, hashtags, or account names for specific searches.

Explore Tab: Highlights trending and popular content.

Filters and Categories: Offers filter options and content categories for browsing.

Suggested Accounts: Recommends accounts based on user interests and engagement.

3.1.1.3 Chat Tab

The Chat/Share tab serves as a dynamic hub for real-time text conversations and multimedia sharing. Users can exchange images, videos, and documents, utilize emojis and stickers for expression, engage in group discussions, search through chat history, manage privacy settings, and receive timely notifications, fostering seamless and personalized communication experiences.



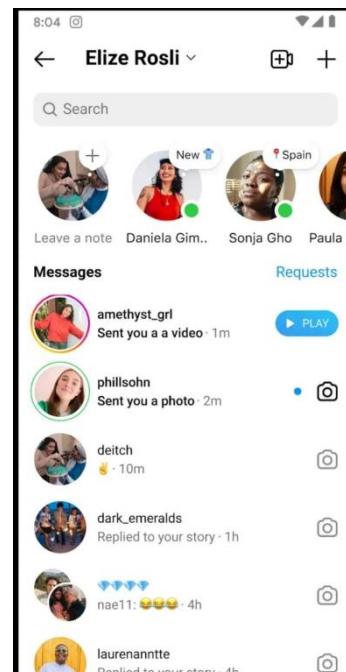
Academic Year: 2023_24



3.1.1.1 Home Page



3.1.1.2 Search Tab



3.1.1.2 Chat Tab

3.1.2 Hardware Interfaces

3.1.2.1 Touchscreen Interface: On mobile devices, such as smartphones and tablets, users interact with the Circlify app through touchscreens. They can navigate through menus, play/pause songs, and manage playlists using touch gestures.

3.1.2.2 Buttons and Controls: On various devices, including smartphones, tablets, and smartwatches, physical buttons and controls (e.g., play, pause, skip, and volume buttons) are used to manage playback without directly interacting with the touchscreen.

3.1.3 Software Interfaces

Operating System APIs

- The Circlify app interacts with operating system APIs on iOS and Android to perform platform-specific tasks like accessing the camera, storing data, and displaying notifications.

Web APIs

- Circlify provides Web APIs that enable third-party applications to integrate with the platform. This includes APIs for user authentication, accessing public profile data, and posting content.

Database APIs

- The app uses database APIs to interact with the SQL and NoSQL databases used on the backend. This includes CRUD operations for objects like users, posts, media, and comments.



Academic Year: 2023_24

Storage APIs

- Storage APIs are used to interface with cloud storage providers for managing user-generated media like images, videos, and files.

Messaging APIs

- Messaging APIs facilitate real-time communication by allowing the exchange of text, media, video calling, and other conversational content between users.

Notification APIs

- Notification APIs on mobile operating systems and web are leveraged to enable push notifications to user devices.

3.1.4 Communications Interfaces

Circlify will employ industry standard protocols and technologies for secure and reliable communication between clients and servers. Data transmission between Circlify apps and API servers will use HTTPS with the latest TLS encryption. Real-time messaging will leverage WebSocket or Socket.io protocols with optimized message serialization formats like Protobuf. For push notifications to devices, platform-specific services like Firebase Cloud Messaging or Apple Push Notification Service will be used.

Where applicable, common lightweight data exchange formats like JSON will be used to maximize interoperability and efficiency. Security measures like encryptions, access controls, and input validation will ensure integrity and privacy of communications. Performance will be optimized through connection pooling, caching, and other best practices.

3.2 Functional Requirements

- User profiles, feeds, and social connections - Allows users to create personal profiles with photos, bios, and interests to represent themselves on the platform and connect with other users they can follow in customizable feeds.
- Sharing of posts with rich text, images, videos, and files - Enables users to make posts that contain formatted text, embedded images, uploaded videos, and attached document files for diverse multi-media sharing.
- Ephemeral stories and vertical video sharing - Offers ephemeral 24-hour stories combining multiple photos/videos with overlays and stickers as well as standalone vertical videos popularized by TikTok.
- Multimedia upload and optimization - Provides functionality to upload images, videos and files while automatically optimizing media for faster loading and streaming on the platform.
- Hashtagging and social discovery - Implements hashtags to tag topics and interests for discovering related content as well as user and content recommendations based on relevance to interests.

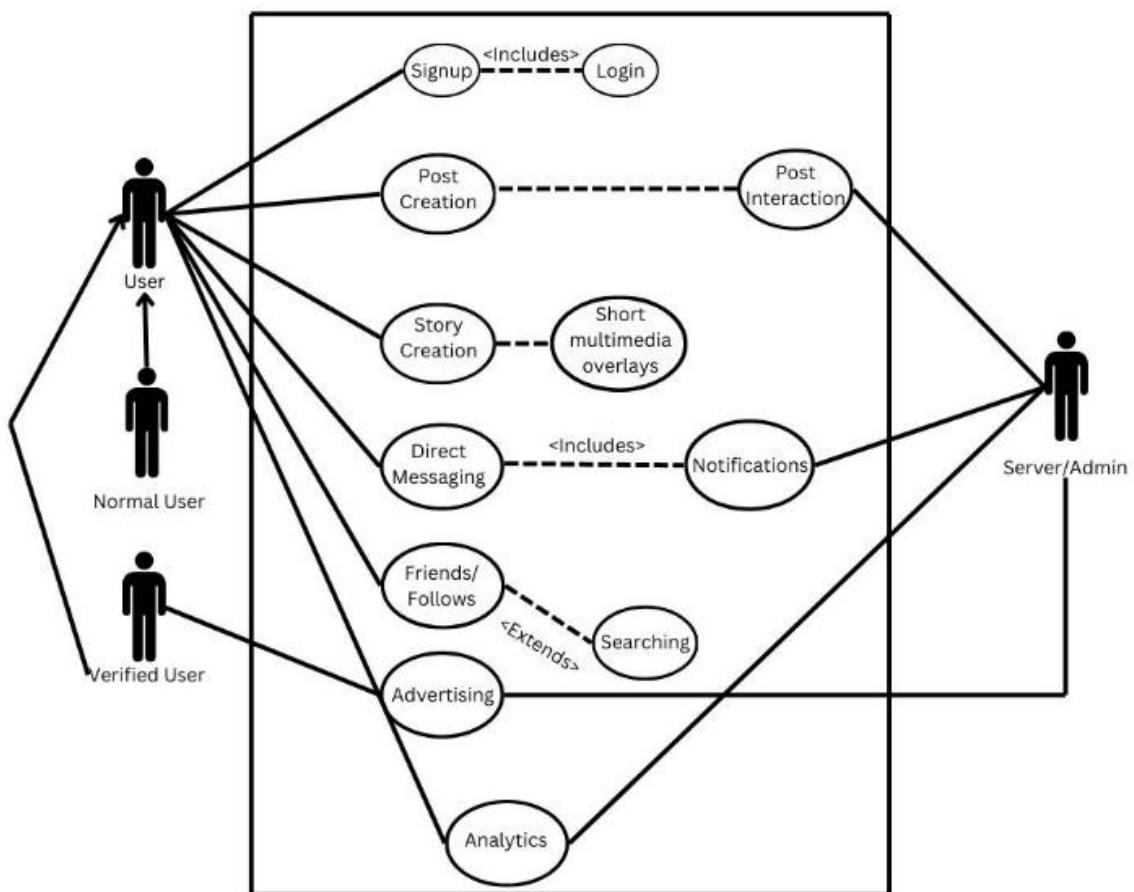


Academic Year: 2023_24

- Business tools for promotions and analytics - Gives business accounts abilities to create targeted promotions to relevant users and view analytics on reach, impressions, and engagement.
- Configurable privacy controls and content moderation - Offers users granular controls to set visibility of posts/profile and provides content flagging along with moderation teams and processes to review inappropriate content.

3.3 Behaviour Requirements

3.3.1 Use Case View





Academic Year: 2023_24

4 Other Non-functional Requirements

4.1 Performance Requirements

- Home feed and search results should load within 2 seconds for good UX
- Media uploads and compressed/optimized within 5 seconds
- Peak load of 100 concurrent users per server
- Push notifications delivered within 3 seconds
- 99.95% uptime for API and database servers
- Automated performance monitoring
- Media processing auto-scaled to maintain performance

4.2 Safety and Security Requirements

- Personally identifiable data encrypted at rest and in transit
- OAuth used for secure third party authentication
- Input validation to prevent code injection vulnerabilities
- Multi-factor authentication required for business accounts
- AI filters to detect inappropriate text, media, hashtags
- Content moderation team reviews flagged posts within 12 hours
- Age gating blocks users under 13 years' old
- Reporting mechanisms for objectionable users/content

4.3 Software Quality Attributes

Circlify will aim to achieve high quality standards guided by specific attributes:

Usability

- The user interfaces will be clean, intuitive, and easy to navigate.
- Onboarding flows will guide new users.
- Navigation and actions will be consistent across platforms.
- Accessibility standards will be followed to support assistive technologies.

To meet usability goals, user research studies and usability testing will be conducted. Feedback will drive iterative design improvements.

Reliability

- Infrastructure will implement redundancy and failover mechanisms.
- Input validation will catch bad data.
- Request throttling will prevent overload.
- Monitoring will enable issue diagnosis and rapid response.
- Disaster recovery provisions and data backup will minimize risk.

Reliability will be measured through uptime, mean time between failures, and ability to recover quickly.



Academic Year: 2023_24

Scalability

- Auto-scaling groups will adjust capacity based on demand.
- Database sharing will distribute load at high volumes.
- Caching and CDNs will reduce server processing.
- Code will be optimized to use resources efficiently.
- Modular services can scale independently.

Scalability will be evaluated under load tests simulating growth.

Maintainability

- Code will adhere to style standards with meaningful naming, comments.
- Version control will track changes and enable rollback.
- Code modularity will isolate components.
- Automated testing will catch regressions quickly.
- Monitoring will provide application insights.
- Developer documentation will capture institutional knowledge.

Maintainability will be measured by change failure rate and mean time to repair.



Academic Year: 2023_24

Appendix B - Group Log

Date	Actors	Work Done
07/02/2024	Amartya Mishra	Analyzed Requirements
07/02/2024	Darshit Sarda	Analyzed Requirements
07/02/2024	Gaurang Bhogle	Analyzed Requirements
07/02/2024	Shubham Mehta	Prepared SRS
14/02/2024	Amartya Mishra	Prepared SRS
14/02/2024	Darshit Sarda	Prepared SRS
14/02/2024	Gaurang Bhogle	Prepared SRS
14/02/2024	Shubham Mehta	Prepared SRS
21/02/2024	Amartya Mishra	Prepared SRS
21/02/2024	Darshit Sarda	Prepared SRS
21/02/2024	Gaurang Bhogle	Prepared SRS
21/02/2024	Shubham Mehta	Prepared SRS