



Continuous Assessment for Laboratory / Assignment sessions

Academic Year 2023-24

Name: Kareena Shah

SAP ID: 60004210243

Course: Software Engineering

Course Code: DJ19CEC601

Year: T.Y. B.Tech.

Sem: VI

Batch: C'32

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	Avg	Avg	Avg	Avg
Course Outcome	1	2	2	2	3	2	5	4	4	6					
1. Knowledge (Factual/Conceptual/Procedural/ Metacognitive)	2	3	3	3	3	3	2	3	2	3		2	2	2	
2. Describe (Factual/Conceptual/Procedural/ Metacognitive)	3	-	3	2	3	3	3	2	3	2		2	2	2	
3. Demonstration (Factual/Conceptual/Procedural/ Metacognitive)	-	-	2	-	2	-	-	-	-	5		2	2	2	
4. Strategy (Analyse & / or Evaluate) (Factual/Conceptual/ Procedural/Metacognitive)	2	2	-	3	3	3	3	3	3	-		2	2	2	
5. Interpret/ Develop (Factual/Conceptual/ Procedural/Metacognitive)	-	3	2	3	-	3	3	3	-	-		-	-	-	
6. Attitude towards learning (receiving, attending, responding, valuing, organizing, characterization by value)	3	3	3	3	3	3	-	-	3	3		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	13	14	13	14	14	14	14	14	14	14	144	10	10	10	
Signature of the faculty member	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	

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

Laboratory marks Σ Avg. = <u>14</u>	Assignment marks Σ Avg. = <u>10</u>	Total Term-work (25) = <u>24</u>
Laboratory Scaled to (15) = <u>14</u>	Assignment Scaled to (10) = <u>10</u>	Sign of the Student: <u>Shreya</u>

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

Shreya
3/5/24

Signature of Head of the Department
Date:

SE Experiment 1.

DATE:

Kareena Shah

60004210243

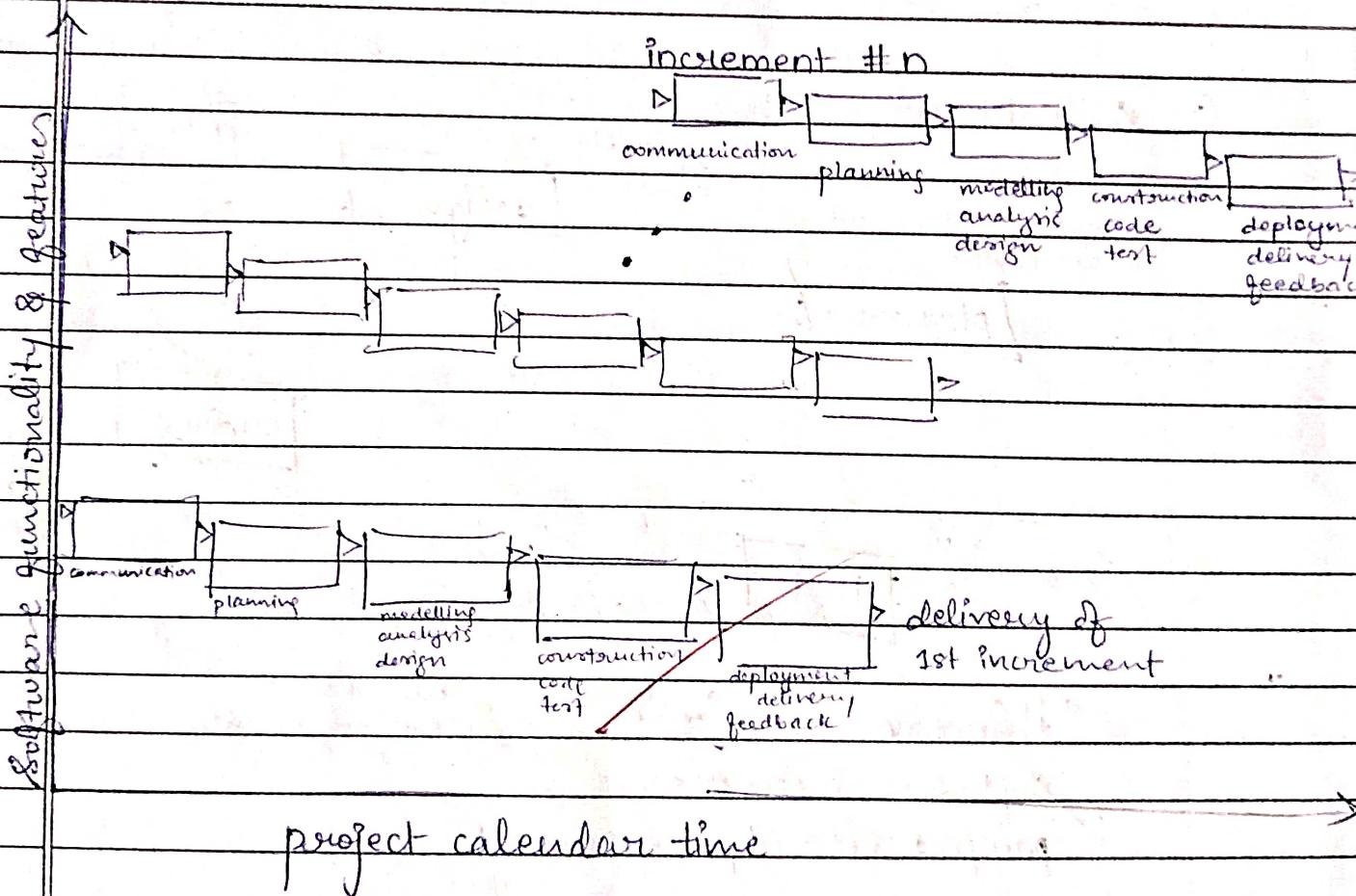
C'32

Aim : To identify suitable life cycle model for your case & justify your model

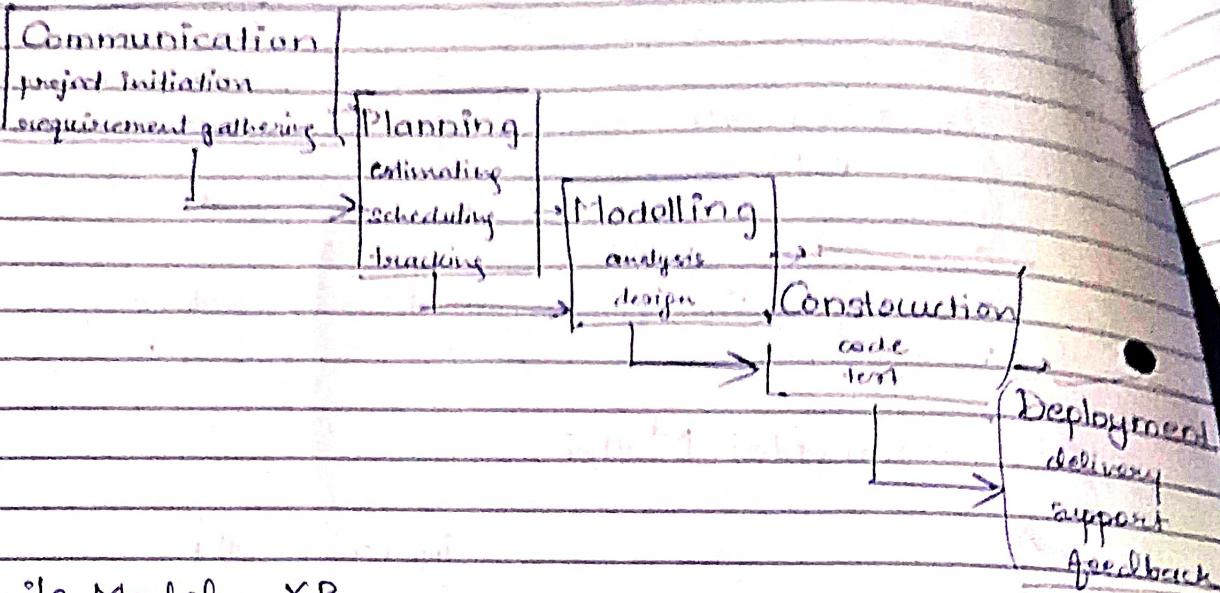
Theory :

A software life cycle model describes entry & exit criteria for each phase.

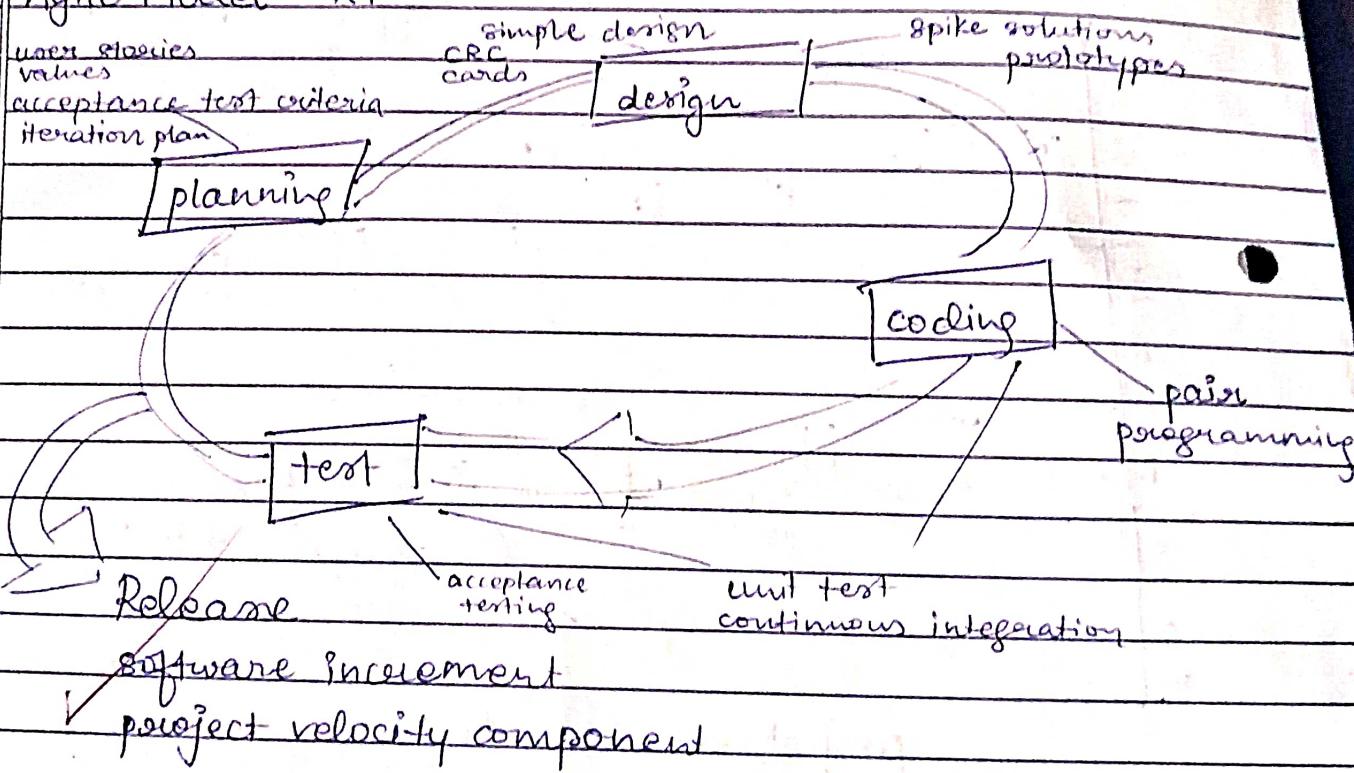
(1) Incremental Model



(2) Waterfall Model



(3) Agile Model - XP



Justification :

In the rapidly evolving landscape of skill assessment, there exists a need for a unified platform that seamlessly integrates resources, practice tests, mock interviews & competitive opportunities.

It caters to a diverse user base including aspiring programmers, professionals in non-technical fields, companies looking to identify & recruit top talent.

Given the evolving nature of educational platforms & the need for rapid adaptation to user feedback, Agile's flexibility & customer-centric approach makes it an ideal choice.

- It allows for iterative development & frequent releases.
- Continuous user feedback ensures that the platform remains aligned with user expectations & industry trends.
- Its emphasis on customer collaboration suits for our application because here user experience is crucial for success.

Conclusion : Thus, we understood different life cycle models & chose the best fitted one for our application.

SE Experiment 3

DATE:

Kareena Shah

60004210243 - 0177

C'32

Aim : Identify scenarios & develop UML Use Case & Class Diagram for the project.

Theory :

→ Students & Working Professionals

(1) Login / Signup

Allows users to create new accounts / log in to the platform using their credentials

(2) Profile Updation

Permits users to update their profiles

(3) Participation in contests

Enables users to participate in coding contests, hackathons & other competitions

(4) Searching Opportunities

Allows users to search & explore job opportunities

For example, platforms like LinkedIn and Glassdoor offer such features.

→ Admin

(1) User Verification

Verify the identity & credentials of users

(2) Conducting Contests

Enables admin to organize & manage coding contests

(3) Validate Solutions

Admin evaluates & validates solutions submitted by participants

(4) Authorization to add opportunities

Admin authorizes & approves the opportunities posted by recruiters

→ Recruiters

- (1) login/signup
- (2) Post Opportunities
- (3) Candidate Selection

Class Diagram

Classes

(1) Coding Questions

This class represents coding questions used in coding contests. Properties like question ID, title, description, difficulty level & list of test cases.

It includes methods to submit solution & validate test cases.

(2) User Actions

This class encapsulates actions a user can perform. Properties like user ID, list of job applications, user's resume, etc.

It includes methods to submit job applications, participate in contests.

(3) User

(4) Contest

(5) Recruiter

(6) Job Application

User Class has 1-1 relationship with User Actions, i.e. each user has a set of actions they can perform.

User Class submits Job Application

Recruiter Class posts Job Application

User Actions class participates in Contest objects

Contest Class includes Coding Questions

User Action class performs actions on Coding Questions

Conclusion : Thus, we created use case & class diagram

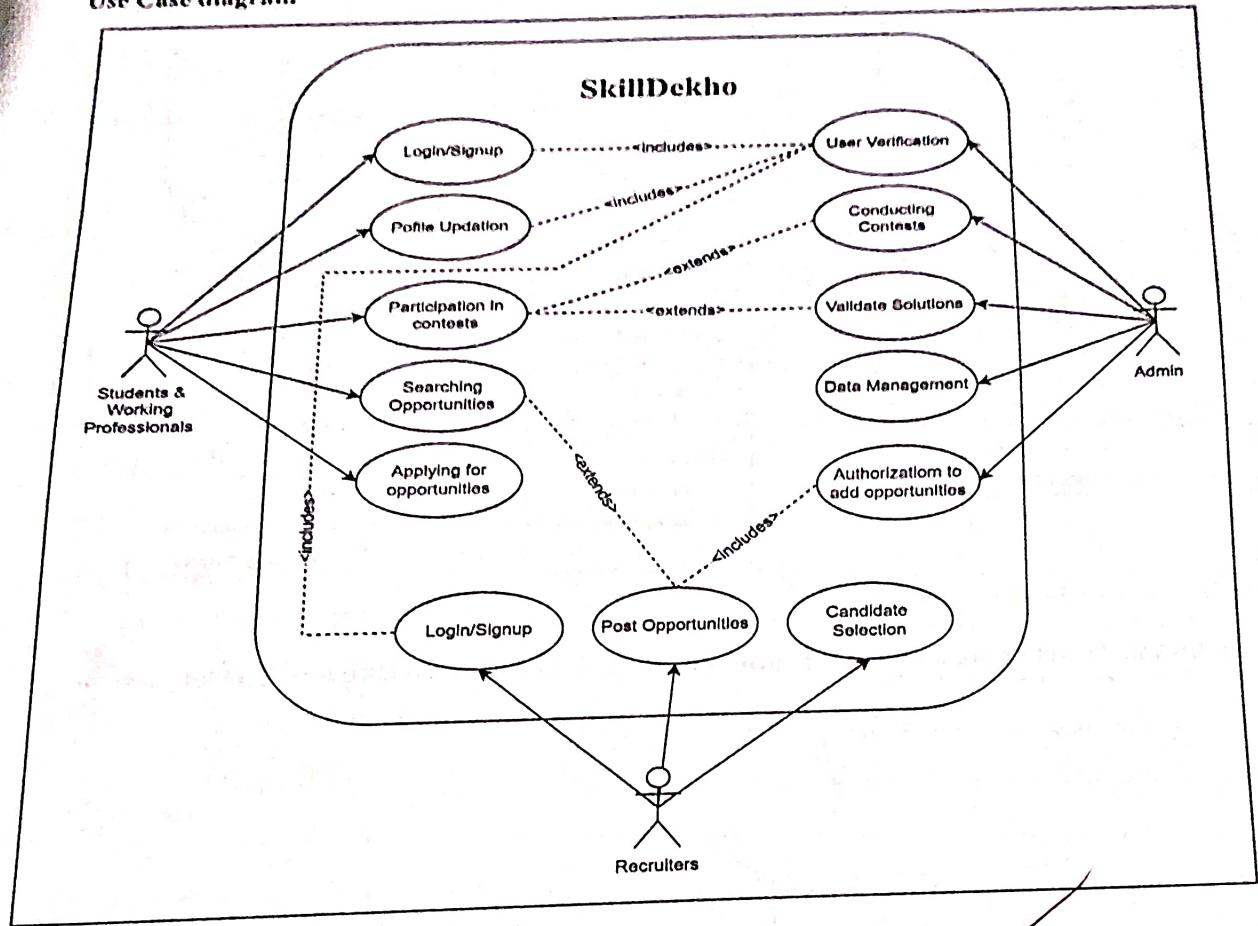
Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
 (Autonomous College Affiliated to the University of Mumbai)
 NAAC Accredited with "A" Grade (CGPA 3.16)



SE Experiment 2

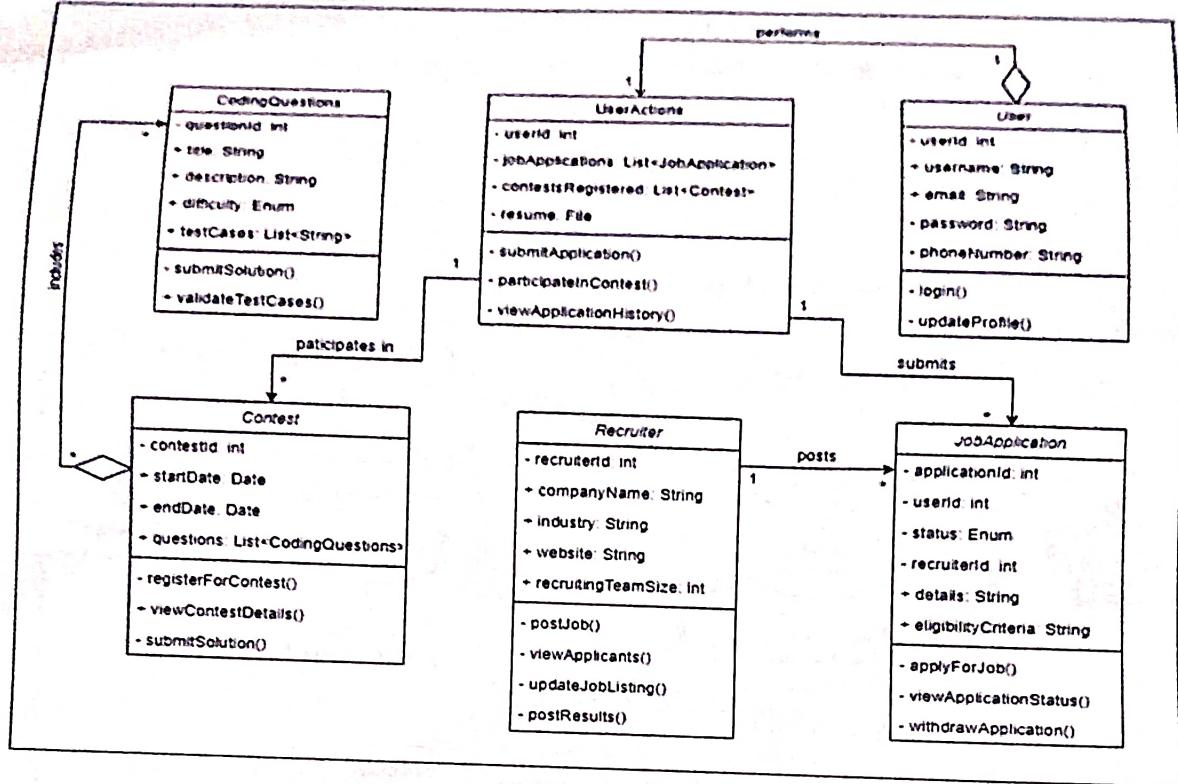
Aim: Identify scenarios & develop UML Use case and Class Diagram for the project

Use Case diagram




**Shri Ville Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**
 (Autonomous College Affiliated to the University of Mumbai)
 NAAC Accredited with "A" Grade (CGPA: 3.18)

Class diagram



Conclusion: Therefore we have implemented use case and class diagram for our project SkillDekho

SE Experiment 4

DATE:

Kreena Shah

60004210243

C'32

Aim : Develop activity diagram & DFD (upto 2 levels) for the project.

Theory :

Activity Diagrams are used to illustrate the flow control in a system & refer to the steps involved in the execution of a use case.

It is a type of behavioral diagram & we can depict both sequential processing & concurrent processing of activities using an activity diagram.

Notations

- Initial State
- Final State
- ↓ Decision
- Node

swimlane

Swimlane

→ Control Flow

Activity State

Activity State

Data Flow Diagram (DFD) represents flow of data of a system. It also gives insight into the inputs & outputs of each entity & the process itself. DFD does not have control flow & no loops for decision making.

Conclusion: Thus, we developed activity diagram & DFD

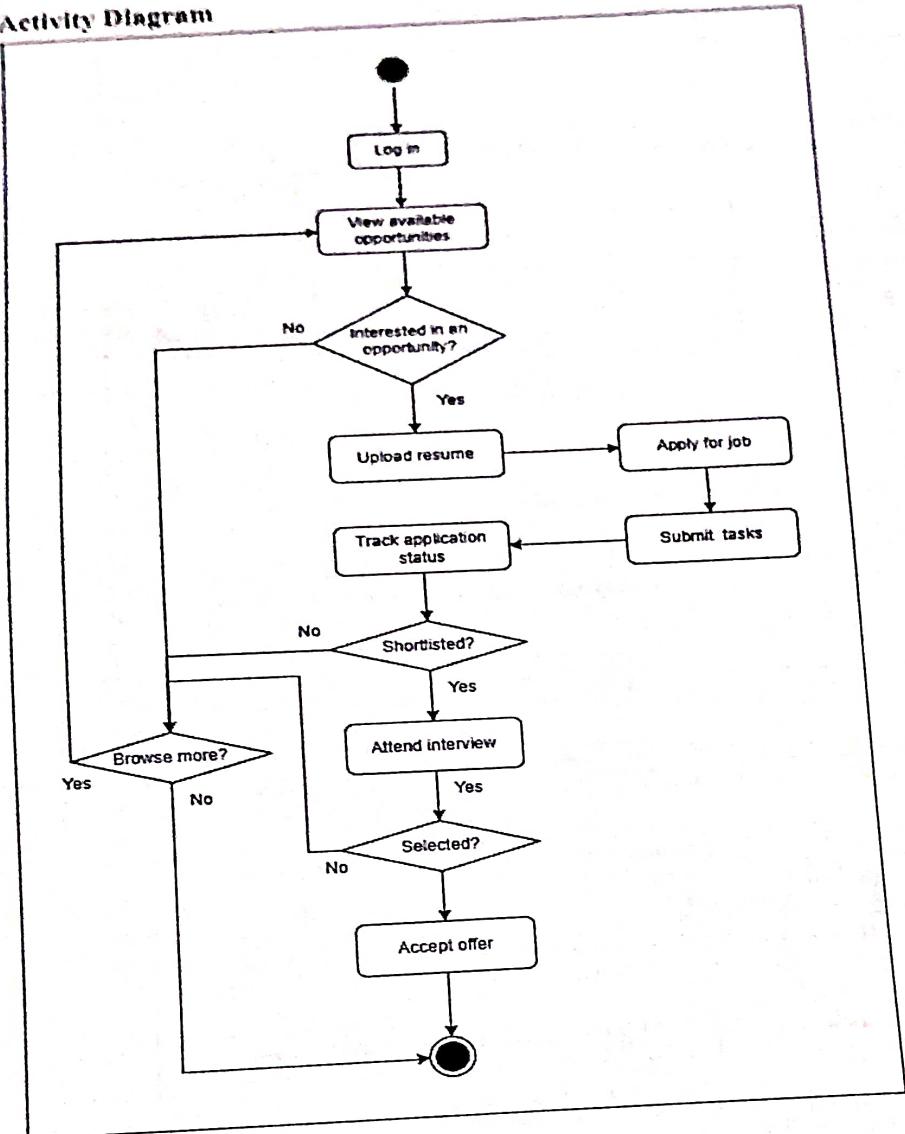
Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA 3.18)



SE Experiment 4

Aim: To implement Activity diagram and DFD

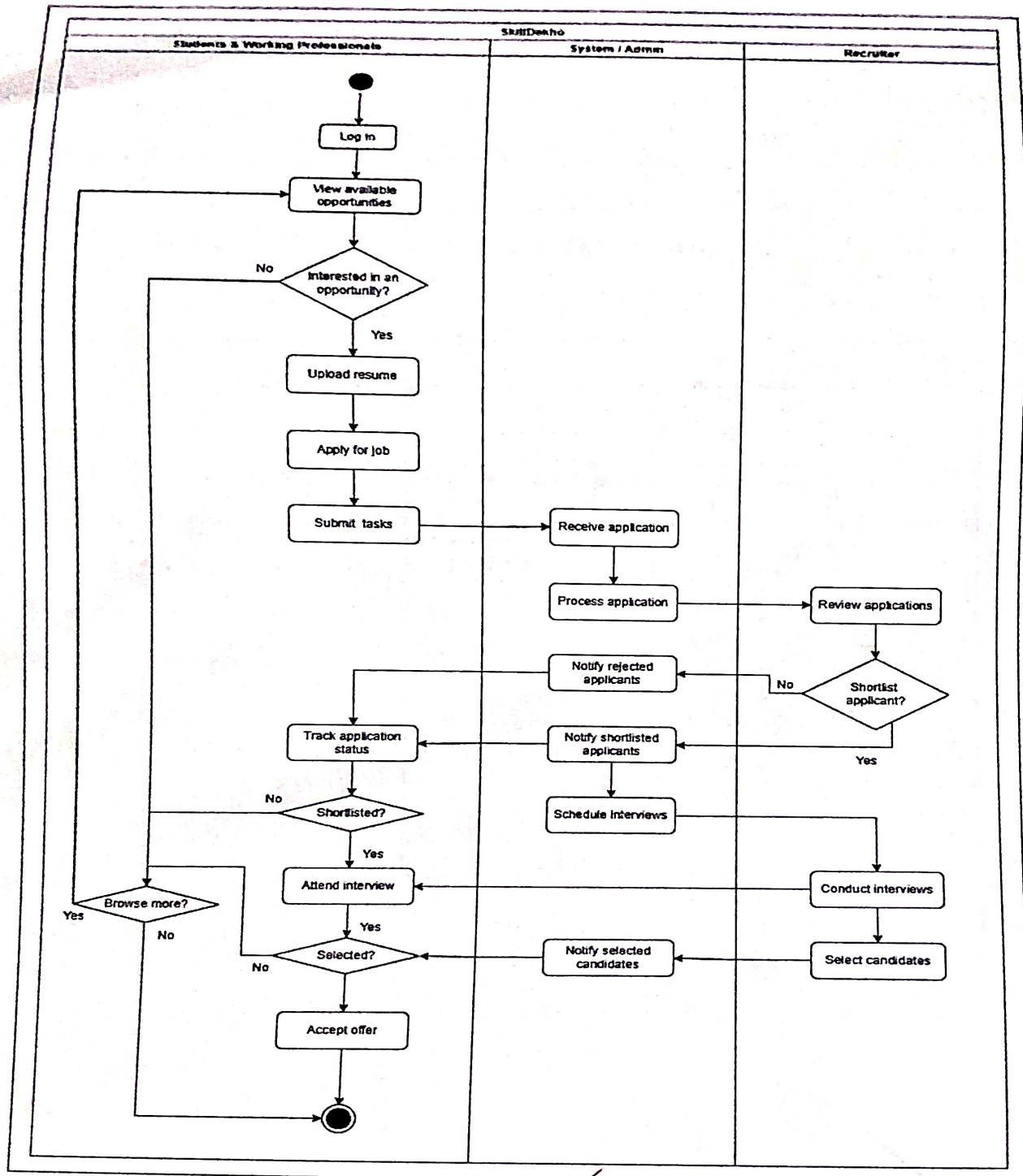
Activity Diagram



Shri Vilas Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
 (Autonomous College Affiliated to the University of Mumbai)
 NAAC Accredited with "A" Grade (CGPA: 3.18)



Swimlane

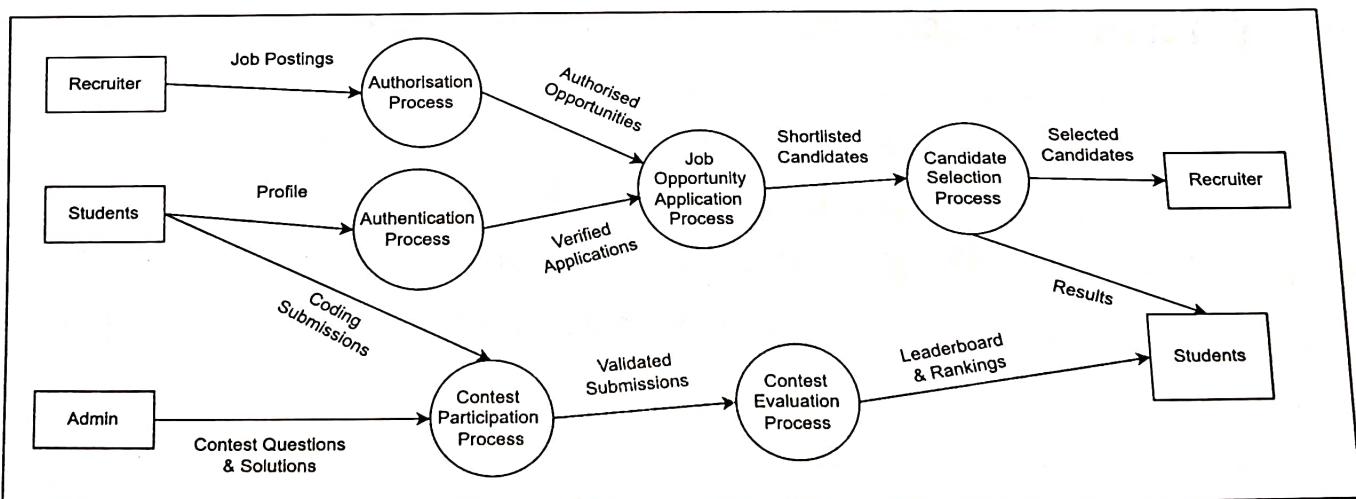
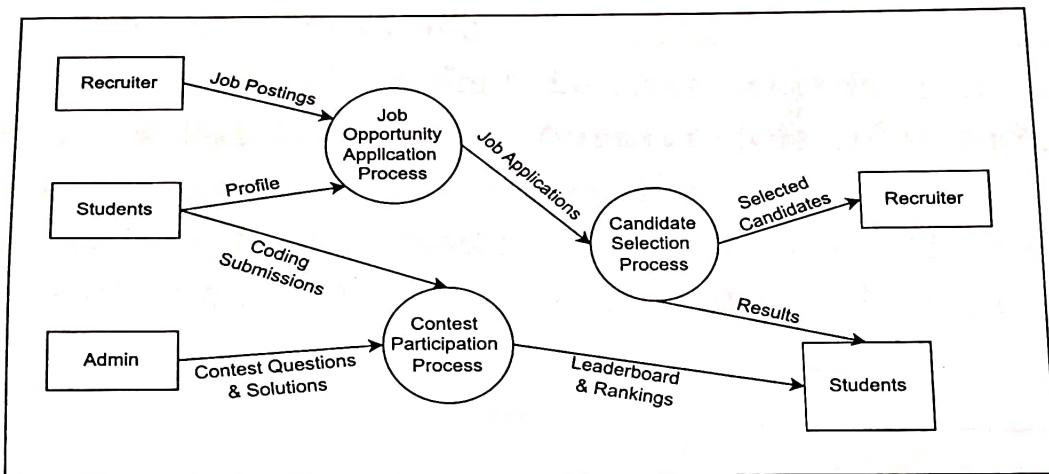
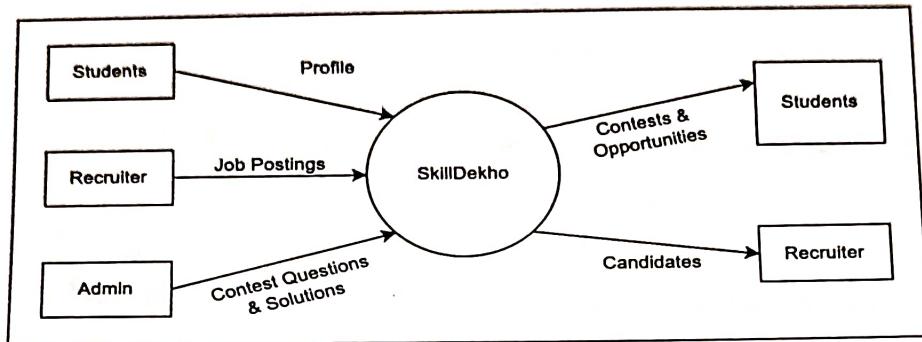




Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA: 3.18)



DFD



Conclusion: Therefore we have implemented Activity diagram and DFD for our project SkillDekho

SE, Experiment 5

DATE:

Kaleena Shah
60004210243
C'32

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

Theory :

Sequence Diagrams

It depicts interactions between objects in a sequential order.
It shows how objects communicate with each other in terms of messages exchanged over time.
Each object is represented as a vertical line & the messages exchanged between them are depicted as horizontal arrows.

Collaboration Diagram

It is also known as communication diagram.
It illustrates the relationships & interactions among objects in the system.
It focuses more on the structural organization of objects & how they collaborate to achieve a specific functionality.
Objects are represented as nodes & the links between them represent messages exchanged.

State Diagram

It represents different states that an object can be in, as well as the transitions between those stages triggered by events. It's particularly useful for modelling the behaviour of objects that can exist in different states & undergo state

DATE:

transitions based on certain conditions.

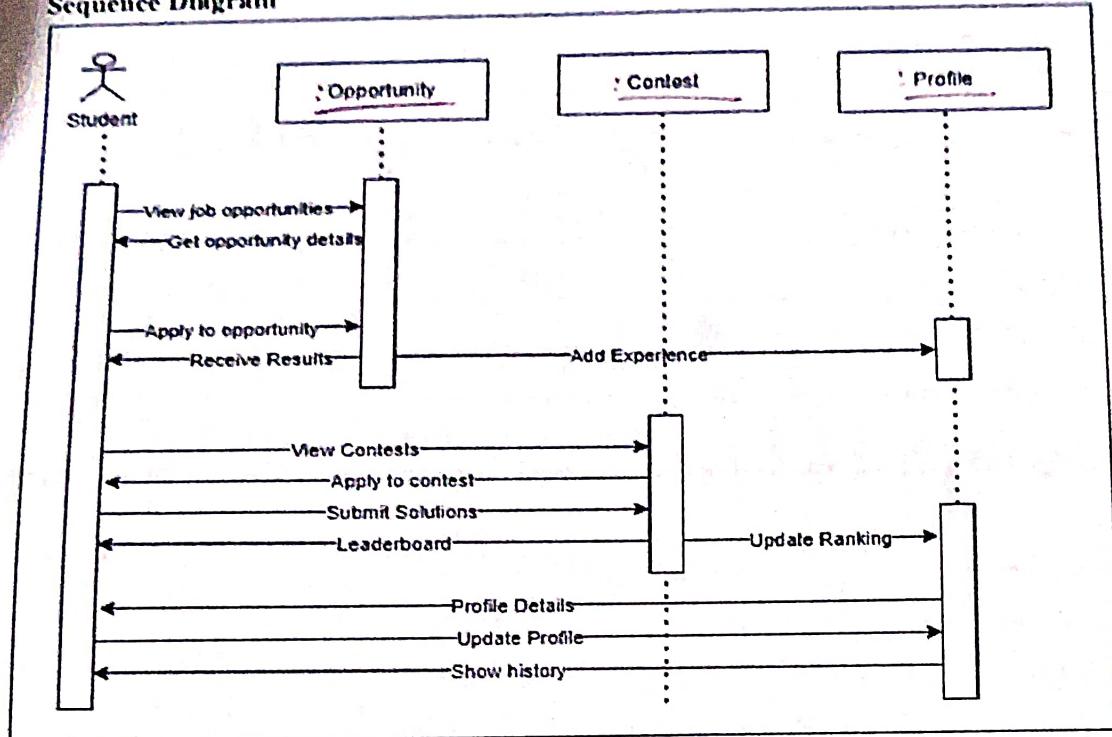
Conclusion: Thus, we created sequence diagrams, collaboration diagrams & state diagrams for our project.



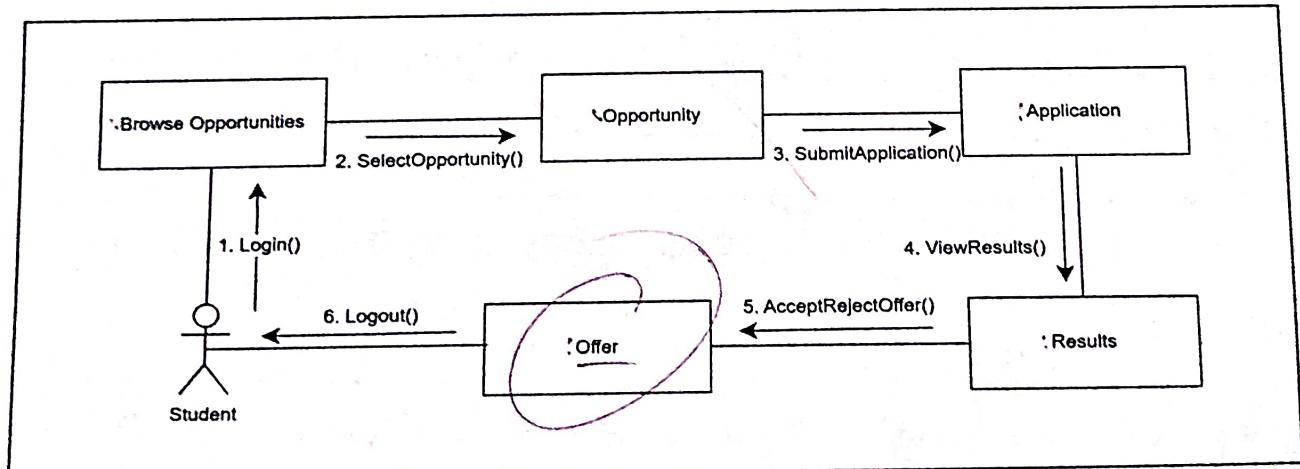
SE Experiment 5

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

Sequence Diagram



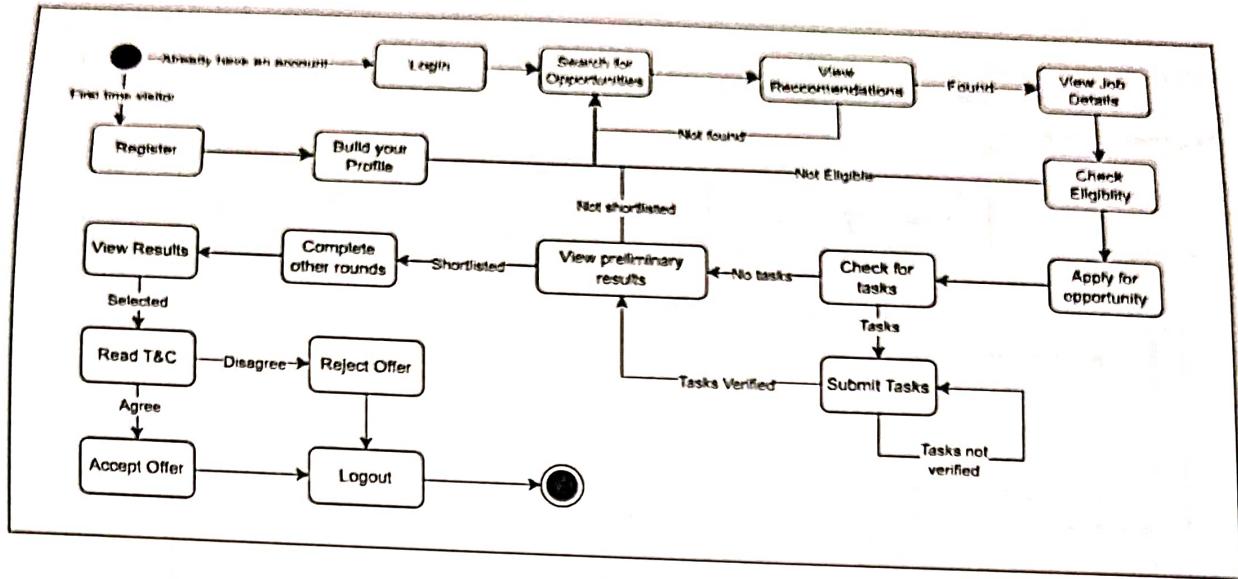
Collaboration




Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
 (Autonomous College Affiliated to the University of Mumbai)
 NAAC Accredited with 'A' Grade (CGPA: 3.18)



State



Conclusion: Therefore we have implemented Sequence, Collaboration and State diagram for our project SkillDekho

S.E. Experiment 6.

DATE:

Kareena Shah
60004B10243
C'32

Aim : Estimate effort & cost required for project using FP/COCOMO.

Create WBS & Gantt chart for the same
use PM tool to depict a plan

Theory :

FP Calculation

Weights

Info Domain Values Count Simple Average Complex

EI	5	3	4	6
EO	4	4	5	7
EQ	3	3	4	6
TIF's	2	7	10	15
ETF's	1	5	7	10

Now, to estimate FP cost, we'll use the assigned domain weights & calculate the Unadjusted Function Points (UAPPF)

$$\begin{aligned} \text{Total UFP} &= 5 \times 4 + 4 \times 5 + 3 \times 3 + 2 \times 7 + 1 \times 5 \\ &= 68 \end{aligned}$$

Now, we use COCOMO to estimate the effort in person months using the calculated UFP

$$\text{Effort} = a * \text{UFP}^b$$

Assuming $a = 2.5, b = 0.38$

$$\text{Effort} = 2.5 * 6.8^{0.38}$$

≈ 41.7 person months

Our project FP is 42

Calculating total person months

$$\text{Total person months} = \frac{\text{Total function points}}{\text{Project Duration}}$$

Average Productivity Per person

Project	Function Points	Effort (Person Months)	Productivity (FP/PM)
P1	10	10	1.0
P2	15	15	1.0
P3	20	20	1.0
P4	25	25	1.0
P5	30	30	1.0
P6	35	35	1.0
P7	40	40	1.0
P8	45	45	1.0
P9	50	50	1.0
P10	55	55	1.0
P11	60	60	1.0
P12	65	65	1.0
P13	70	70	1.0
P14	75	75	1.0
P15	80	80	1.0
P16	85	85	1.0
P17	90	90	1.0
P18	95	95	1.0
P19	100	100	1.0
P20	105	105	1.0
P21	110	110	1.0
P22	115	115	1.0
P23	120	120	1.0
P24	125	125	1.0
P25	130	130	1.0
P26	135	135	1.0
P27	140	140	1.0
P28	145	145	1.0
P29	150	150	1.0
P30	155	155	1.0
P31	160	160	1.0
P32	165	165	1.0
P33	170	170	1.0
P34	175	175	1.0
P35	180	180	1.0
P36	185	185	1.0
P37	190	190	1.0
P38	195	195	1.0
P39	200	200	1.0
P40	205	205	1.0
P41	210	210	1.0
P42	215	215	1.0
P43	220	220	1.0
P44	225	225	1.0
P45	230	230	1.0
P46	235	235	1.0
P47	240	240	1.0
P48	245	245	1.0
P49	250	250	1.0
P50	255	255	1.0
P51	260	260	1.0
P52	265	265	1.0
P53	270	270	1.0
P54	275	275	1.0
P55	280	280	1.0
P56	285	285	1.0
P57	290	290	1.0
P58	295	295	1.0
P59	300	300	1.0
P60	305	305	1.0
P61	310	310	1.0
P62	315	315	1.0
P63	320	320	1.0
P64	325	325	1.0
P65	330	330	1.0
P66	335	335	1.0
P67	340	340	1.0
P68	345	345	1.0
P69	350	350	1.0
P70	355	355	1.0
P71	360	360	1.0
P72	365	365	1.0
P73	370	370	1.0
P74	375	375	1.0
P75	380	380	1.0
P76	385	385	1.0
P77	390	390	1.0
P78	395	395	1.0
P79	400	400	1.0
P80	405	405	1.0
P81	410	410	1.0
P82	415	415	1.0
P83	420	420	1.0
P84	425	425	1.0
P85	430	430	1.0
P86	435	435	1.0
P87	440	440	1.0
P88	445	445	1.0
P89	450	450	1.0
P90	455	455	1.0
P91	460	460	1.0
P92	465	465	1.0
P93	470	470	1.0
P94	475	475	1.0
P95	480	480	1.0
P96	485	485	1.0
P97	490	490	1.0
P98	495	495	1.0
P99	500	500	1.0
P100	505	505	1.0
P101	510	510	1.0
P102	515	515	1.0
P103	520	520	1.0
P104	525	525	1.0
P105	530	530	1.0
P106	535	535	1.0
P107	540	540	1.0
P108	545	545	1.0
P109	550	550	1.0
P110	555	555	1.0
P111	560	560	1.0
P112	565	565	1.0
P113	570	570	1.0
P114	575	575	1.0
P115	580	580	1.0
P116	585	585	1.0
P117	590	590	1.0
P118	595	595	1.0
P119	600	600	1.0
P120	605	605	1.0
P121	610	610	1.0
P122	615	615	1.0
P123	620	620	1.0
P124	625	625	1.0
P125	630	630	1.0
P126	635	635	1.0
P127	640	640	1.0
P128	645	645	1.0
P129	650	650	1.0
P130	655	655	1.0
P131	660	660	1.0
P132	665	665	1.0
P133	670	670	1.0
P134	675	675	1.0
P135	680	680	1.0
P136	685	685	1.0
P137	690	690	1.0
P138	695	695	1.0
P139	700	700	1.0
P140	705	705	1.0
P141	710	710	1.0
P142	715	715	1.0
P143	720	720	1.0
P144	725	725	1.0
P145	730	730	1.0
P146	735	735	1.0
P147	740	740	1.0
P148	745	745	1.0
P149	750	750	1.0
P150	755	755	1.0
P151	760	760	1.0
P152	765	765	1.0
P153	770	770	1.0
P154	775	775	1.0
P155	780	780	1.0
P156	785	785	1.0
P157	790	790	1.0
P158	795	795	1.0
P159	800	800	1.0
P160	805	805	1.0
P161	810	810	1.0
P162	815	815	1.0
P163	820	820	1.0
P164	825	825	1.0
P165	830	830	1.0
P166	835	835	1.0
P167	840	840	1.0
P168	845	845	1.0
P169	850	850	1.0
P170	855	855	1.0
P171	860	860	1.0
P172	865	865	1.0
P173	870	870	1.0
P174	875	875	1.0
P175	880	880	1.0
P176	885	885	1.0
P177	890	890	1.0
P178	895	895	1.0
P179	900	900	1.0
P180	905	905	1.0
P181	910	910	1.0
P182	915	915	1.0
P183	920	920	1.0
P184	925	925	1.0
P185	930	930	1.0
P186	935	935	1.0
P187	940	940	1.0
P188	945	945	1.0
P189	950	950	1.0
P190	955	955	1.0
P191	960	960	1.0
P192	965	965	1.0
P193	970	970	1.0
P194	975	975	1.0
P195	980	980	1.0
P196	985	985	1.0
P197	990	990	1.0
P198	995	995	1.0
P199	1000	1000	1.0



SE Experiment 6

Aim: Estimate effort and cost required using FP/COCOMO for the project. Create WBS and Gantt Chart for the same.

FP Cost Estimation

Information Domain Value	Count	Simple	Average	Complex
External Inputs	5	3	4	6
External Outputs	4	4	5	7
External Inquiries	3	3	4	6
Internal User Interface	2	7	10	15
External User Interface	1	5	7	10

External Inputs (EI):

Count: 5

- User registration
- User login
- Profile update
- Job opportunities
- Eligibility criteria

Complexity Scale: Average

External Outputs (EO):

Count: 4

- Search results display
- Leaderboard and ranking
- Profile view
- Job recommendation

Complexity Scale: Average

External Inquiries (EQ):

Count: 3:

- Job filters


**Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA: 3.18)



- Candidate filters
- Contest Problem types

Complexity Scale: Simple

Internal Logic Files (ILF):

Count: 2

- User profile data
- Job database

Complexity Scale: Average

External Interface Files (EIF):

Count: 1

- Integration with external authentication system

Complexity Scale: Average

$$\begin{aligned}\text{Total UFP} &= (\text{EI} * 4) + (\text{EO} * 5) + (\text{EQ} * 3) + (\text{ILF} * 7) + (\text{EIF} * 5) \\ &= (5 * 4) + (4 * 5) + (3 * 3) + (2 * 7) + (1 * 5) \\ &= 20 + 20 + 9 + 14 + 5 \\ &= 68\end{aligned}$$

$$\text{Effort} = a * (\text{UFP})^{(b)}$$

Let's assume typical values for a and b: $a = 2.5$, $b = 0.38$

$$\text{Effort} = 2.5 * (68)^{(0.38)}$$

$$\approx 41.7 \text{ person-months}$$

To calculate the total person-months and average productivity per person,

Team Size: 8 developers

Project Duration: 7 months

$$\text{Total Person-Months} = \text{Effort} / \text{Project Duration}$$

$$= 41.7 \text{ person-months} / 7 \text{ months}$$

$$\approx 5.957 \text{ person-months per month}$$

$$\text{Average Productivity per Person} = \text{Total Person-Months} / \text{Team Size}$$

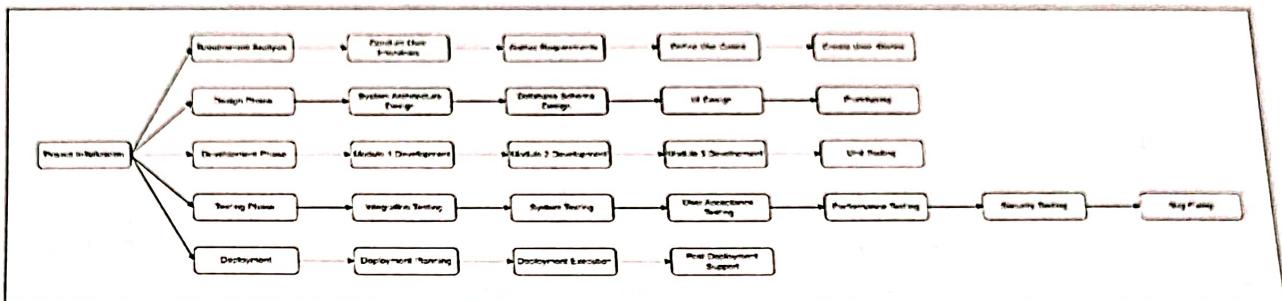
$$\approx 5.957 \text{ person-months per month} / 8 \text{ developers}$$

$$\approx 0.744 \text{ person-months per developer per month}$$

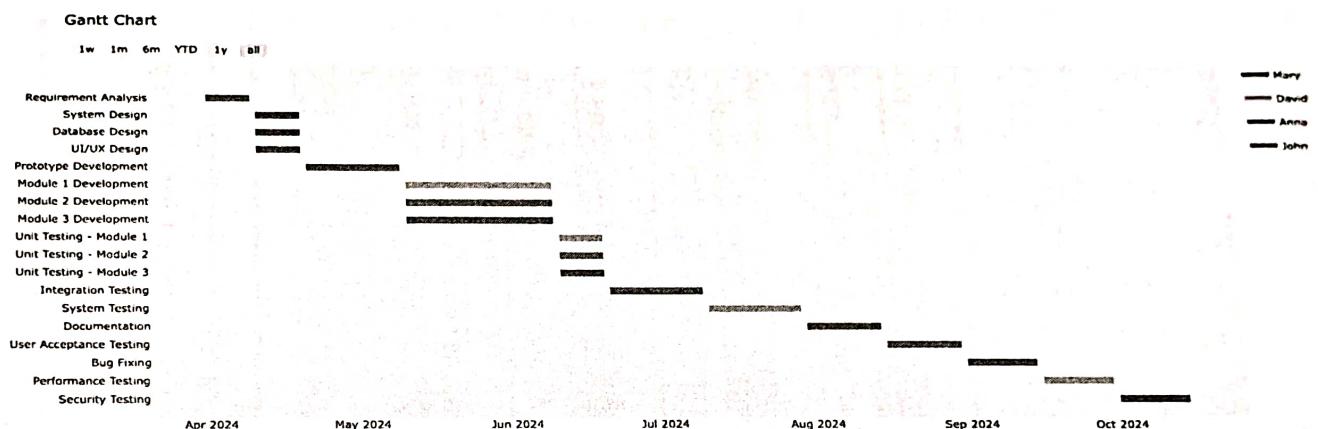
Shri Ville Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
 (Autonomous College Affiliated to the University of Mumbai)
 NAAC Accredited with "A" Grade (CGPA: 3.18)



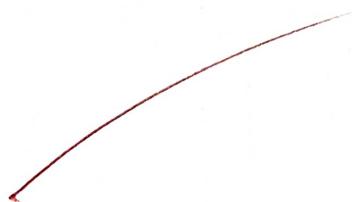
Work Break Down



Gantt Chart



Conclusion: Therefore we have implemented the effort and cost required using FP/COCOMO for the project. Create WBS and Gantt Chart for our project SkillDekho



Project Name:	SkillDekho - Educational & Employment Platform
Reference Document:	SkillDekho SRS Document
Created by:	
Date of creation:	13-Feb-24
Date of review:	29-Mar-24

Test scenario ID	Requirement-document index	Test scenario description	Importance	No. of test cases
TS_01	Section 3, Page 3	Ensure smooth registration and login processes for users.	High	3
TS_02	3.1.1	Ensure users can access coding challenges seamlessly.	Medium	2
TS_03	3.1.2	Validate the process of submitting coding solutions and receiving feedback.	High	3
TS_04	3.1.3	Ensure seamless hosting and participation in coding contests.	Medium	2
TS_05	3.1.3	Validate the retrieval of contest results and participant data.	High	2
TS_04	3.1.4	Validate the process of registering for recruitment events.	High	2
TS_04	3.1.4	Ensure smooth conduct of technical assessments during recruitment events.	Medium	2
TS_04	3.1.5	Ensure users can access learning resources through the platform's learning management system (LMS).	Medium	3
TS_04	3.1.5	Ensure platform compatibility across devices and browsers	Low	2
TS_05	3.1.5	and evaluate		

Project Name	SkillDekho - Educational & Employment Platform
Reference Document	SkillDekho SRS Document
Created by	
Date of creation	13-Feb-24
Date of review	29-Mar-24

Test case ID	Test Objective	Precondition	Steps:	Test data	Expected result
TC_MI_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. Navigate to the login page. 2.In the login Panel, enter the username	"A valid username"	The user is logged in successfully.
TC_MI_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"	
TC_MI_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
			3. Click "Login" button		
TC_MI_2.1	Verify that a user can attempt a coding problem.	1.The user is logged in.2.The problem set is available.	1. Navigate to the "Problems" section. 2.Select & read the problem statement and constraints. 3.Write the code solution in the provided editor. 4.Click the "Submit" button.	Code solution for the selected problem.	The user's solution is evaluated, and the result is displayed.
TC_MI_2.2	Verify that a user can participate in a coding contest.	The user is logged in. An ongoing coding contest is available.	1.Navigate to the "Contests" section. 2.Select an ongoing contest from the list. 3.Read the contest rules and problem statements. 4.Attempt the contest problems within the given time limit.	Solutions for contest problems within the time limit.	The user's solutions are evaluated, and the contest

			4. Submit your solution and wait for feedback.		
TC_MI_2.3	Verify that a user can view learning resources.	The user is logged in. Learning resources are available.	1. Navigate to the "Learning" section. 2. Browse through the available courses and tutorials. 3. Select & view a course or tutorial.		The selected learning resource is displayed, and the user can access that content.
TC_MI_3.1	To verify that users receive detailed feedback on their submitted coding solutions.	The user has submitted a coding solution, and the evaluation process is complete.	1. Navigate to the "Problems" section. 2. Select a previously attempted problem. 3. Review the submitted solution. 4. Click the "Get Feedback" button.	Submitted coding solution	The user receives detailed feedback on their solution, including comments, suggestions, and
TC_MI_3.2	To ensure users can track their progress and view relevant statistics.	The user is logged in. They have attempted coding problems and participated in contests.	1. Navigate to the user's profile or dashboard. 2. View the "Progress" section.	User's activity and participation data.	The user can see their progress, including the number of problems solved, contests participated in, ranking, and other relevant statistics.
TC_MI_3.3	To verify that users can access contest results and participant data.	The user has participated in a coding contest, and the contest results are available.	1. Navigate to the "Contests" section. 2. Select the completed contest. 3. View the contest results and participant data.	Contest results and participant data.	The user can access the contest results, including participant rankings.
TC_MI_4.1	To confirm that administrators can successfully host coding contests on the platform.	The user is logged in as an administrator, and the platform allows users to host contests.	1. Navigate to the "Contests" section 2. Click on the "Host Contest" button. 3. Provide contest details (name, duration, problem set, rules, etc.). 4. Schedule the contest date and time. 5. Review and submit the contest details.	Contest details	The coding contest is successfully created and scheduled.
TC_MI_4.2	To ensure users can retrieve contest results and participant data after a contest has ended.	The user is logged in, and they have participated in a coding contest whose results are available.	1. Navigate to the "Contests" section.	Contest results and participant data.	The user can access the contest results, including participant rankings,

			2.Select the completed contest.		
			3.View the contest results and participant data.		
TC_MI_4.3	To verify that users understand and acknowledge contest rules before participating.	The user is logged in and intends to participate in a coding contest for which the rules are available.	1.Navigate to the "Contests" section 2.Select the upcoming contest.	Contest rules and guidelines.	The user understands and acknowledges the contest rules before.
			3.Read and verify the contest rules and guidelines.		
TC_MI_5.1	To confirm that the platform supports and evaluates coding solutions correctly for multiple	The user is logged in, and coding problems for different programming languages	1.Navigate to the "Problems" section. 2.Select a problem for a specific programming 3.Attempt the problem using the selected language. 4.Submit the solution	Coding problem, programming language selection, submitted solution.	The user's solution is evaluated correctly, regardless of the
TC_MI_5.2	To verify that the platform is compatible with different devices and browsers.	The platform is accessible via various devices and browsers.	1.Access the platform using different devices 2.Access the platform using different browsers	Various devices and browsers.	The platform displays and functions correctly across different devices and browsers,

SE Experiment 8

DATE:

Kareena Shah
60004210243
C'82

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

Theory :

Risk	Category	Probability	Impact
Underestimating necessary database size	PS	80%	
Lack of development experience	TI	50%	
Technology not meeting requirements	TI	20%	
Lack of user adoption due to poor marketing / user experience	BU	30%	
Poor task management	PT	45%	
Absence of clear monetization strategy	PT	45%	
Absence of a structured change control process	BU	40%	
Poor quality documentation	PI	75%	

TI : Technical Issue Risk

BU : Business Risk

PS : Product size risk

PI : Process Risk

CR : customer related risk

DATE:

RISK for RMMM

Lack of user adoption due to poor marketing or user experience.

Risk Exposure :

Impact: Moderate

Probability: Low


Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
 NAAC Accredited with "A" Grade (CGPA: 3.18)

SE Experiment 8

Aim: Design test RMMM plan.

Theory:

RMMM

Risk Information sheet			
Risk ID: P02-4-32	Date: 02-05-2024	Prob: 70%	Impact: Critical
Description: Lack of user adoption due to poor user experience or lack of engaging content on the SkillsDekho platform.			
Refinement/context:			
<ol style="list-style-type: none"> Subcondition1: Unintuitive user interface or navigation making it difficult for users to access coding problems and learning resources. Subcondition2: Lack of diverse and engaging coding problems or learning content. Subcondition3: Poor performance or frequent technical issues on the platform. 			
Mitigation/monitoring:			
<ol style="list-style-type: none"> Conduct usability testing with target users and gather feedback to improve the user experience. Collaborate with educational institutions and industry experts to curate high-quality coding problems and learning resources. Implement robust performance monitoring and issue tracking systems to identify and resolve technical issues promptly. 			
Management/contingency plan/trigger:			
<ol style="list-style-type: none"> Allocate a budget of INR 3,00,000 for user experience (UX) design and usability improvements. Partner with educational content providers or hire subject matter experts to enhance the learning resources (Estimated cost: INR 5,00,000 per year). Implement load testing and performance optimization strategies (Estimated cost: INR 2,00,000). 			
Current Status - Requires ongoing monitoring and mitigation strategies.			
Originator: Kapil Kashyap	Assigned To: Kreena Shah		

Conclusion: Thus we have implemented RMMM for our project SkillDekho

SE Experiment 9

DATE:

Kaleena Shah

80004210243

C'22

Aim : Study of Configuration Management using Github.

Theory :

Configuration Management refers to the process of systematically managing changes to a system's configuration throughout its lifecycle. This encompasses

- identification
- control
- tracking of configuration items

which can include source code, documentation, dependencies & other artifacts.

Configuration Management involves :

- (1) Version Control
- (2) Change Management
- (3) Baseline Management
- (4) Release Management

In context of SE, Github serves as a configuration management tool

(1) Version Control

(2) Branching & Merging

(3) Code Reviews

(4) Issue Tracking

(5) Collaboration

(6) Documentation.

Conclusion : Thus, we understood configuration management & implemented on github.

QUESTION & ANSWER ON CONFIGURATION MANAGEMENT

- a) To interact with the hardware, framework needs to be connected to the hardware after making any changes in the hardware, it is required to make changes in the software which is difficult.

Configuration management is used to reduce the time taken to make changes in the hardware.

- b) Evolution of configuration management
 - Configuration Management
 - Configuration Management
 - Configuration Management
 - Configuration Management

c) Configuration management is the process of maintaining

d) Configuration management is the process of maintaining

e) Configuration management

f) Configuration management is the process of maintaining

g) Configuration management

h) Configuration management

i) Configuration management

j) Configuration management

k) Configuration management



Shri Ville Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA: 3.18)



SE Experiment 9

Aim: Study of Configuration Management using GitHub
Theory:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
Ps C:\Users\Admin\OneDrive\Desktop> git clone https://github.com/Viveknachak/SE-SkillLab9.git
Cloning into 'SE-SkillLab9'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100%, 3/3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (3/3), done.
```

```
PS C:\Users\Admin\OneDrive\Desktop> git clone https://github.com/Viveknachak/SE-SkillLab9.git
Cloning into 'SE-SkillLab9'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100%, 3/3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (3/3), done.
```

```
PS C:\Users\Admin\OneDrive\Desktop> git clone https://github.com/Viveknachak/SE-SkillLab9.git
Cloning into 'SE-SkillLab9'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100%, 3/3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (3/3), done.
```


**Shri Ville Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA: 3.18)



```
git clone https://github.com/KreenaShah/SE-SkillDekho.git
cd SE-SkillDekho
git pull
git branch -r --set-upstream-to=origin/main
git checkout main
git merge -s ours origin/main
git push -u origin main
git push --tags
```

changes made in similar line #2

KreenaShah merged 1 commit into master from ksd11 now

Conversation 0 Commits 1 Checks 0 Files changed 1

KreenaShah commented 5 minutes ago

No description provided

1 changes

KreenaShah merged commit e23f6d4 into master now

Pull request successfully merged and closed

You're all set—the ksd11 branch can be safely deleted.

Add a comment

Write Preview

Add your comment here...

Owner: fad5423

Reviewers: No reviews

Assignees: No one assigned

Labels: None yet

Projects: None yet

Milestone: None yet

Development: English (United States) English (India)

To switch input methods, press Windows key + space

Conclusion: Thus we have implemented Configuration Management using GitHub

SE Experiment 10

DATE:

Kareena Shah
60004210243
C'32

Aim : To understand DevOps.

Theory :

DevOps is a set of practices that combines software development (Dev) & IT operations (ops).

It aims to shorten the systems development life cycle & provide continuous delivery with high software quality.

DevOps emphasizes collaboration, automation & integration between development & operations team to improve speed & quality of delivering software applications & services.

Tools used in different DevOps stages

(1) Plan

- Jira
- Trello
- Asana

(2) Code

- Git, GitHub
- Bitbucket

(3) Build

- Jenkins
- Travis CI

(4) Test

- Selenium
- Postman

(5) Deployment

- Docker
- Kubernetes

(6) Operate

- Nagios
- ELK Stack

(7) Monitor

- Datadog
- Grafana

(8) Feedback

- Slack
- MS teams.

Conclusion : Thus, we understood devops & various tools are explored.

SE Assignment 1

DATES:

Kreana Shah

60004210243 - 0133

C'32

- (1) Elaborate the task set for creating component level Design in OO projects
→ The task set for creating component level design in OO project
 - (1) Identify all designs that correspond to the problem domain.
 - (2) Identify all design classes that correspond to the infrastructure domain
 - (3) Elaborate all design classes that are not acquired as reusable components
 - (i) Specify message details when classes / components collaborate
 - (ii) Identify appropriate interfaces for each component
 - (iii) Elaborate attributes & define datatypes & data structures required to implement them
 - (iv) Describe processing flow within each operation in detail
 - (4) Describe persistent data sources & identify the classes required to manage them
 - (5) Develop & elaborate behavioural representations for a class component
 - (6) Elaborate deployment diagrams to provide additional implementation details
 - (7) Factor every component level design representation & always consider alternatives

(2) The golden rules of user interface are :

- (1) Visibility
- (2) Feedback
- (3) Consistency
- (4) Flexibility
- (5) Simplicity

Methodology

(a) Place user in control

(b) Reduce user's memory load

(c) Make interface consistent

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

Provide for flexibility in interaction

Streamline interaction as skill level advances & allow interaction to be customized

Hide technical internals from casual user

Design for direct interaction with objects that appear on screen

(d) Reduce user's memory load

Reduce demand on short term memory

Establish meaningful defaults

Define shortcuts, intuitive

Disclose information in progressive fashion

(e) Make interface consistent

Allow user to put the current task into a meaningful context

(3) Transform mapping is a process used in software engineering to map the data flow from the source to destination.

It is a set of design steps that allows a DFD with transform flow characteristics to be mapped into a specified architecture style.

To map DFD into software architecture, we would initiate the following design steps :

- (1) Review the fundamental system model
- (2) Review & refine data flow diagrams for software
- (3) Determine whether the DFD has transformer transaction flow characteristics
- (4) Isolate the transform centre by specifying incoming & outgoing flow boundaries
- (5) Perform 1st level factoring
- (6) Perform 2nd level factoring
- (7) Refine 1st 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.

Steps :

- (1) Identify logical transactions
- (2) Each logical transaction is mapped to a series of database operations
- (3) Transaction mapping ensures automatic transaction.

DATES:

- ④ It involves optimizing performance of transactions.

SE Assignment 2

DATE:

Kareena Singh

0000112102413 - 0179

C'92

- (1) A DevOps Tool Chain refers to a set of tools, technologies & processes used to facilitate & automate the various stages of software development, delivery & operations within a DevOps environment.

It encompasses a wide range of tools that helps teams collaborate, automate repetitive tasks, ensure consistency, & accelerate the delivery of high quality software.

As mentioned above, different tools are used at different stages.

They are as follows :

- (1) Collaboration
- (2) Planning
- (3) Source Control
- (4) Issue Tracking
- (5) Configuration Management
- (6) Continuous Integration
- (7) Binary Repositories
- (8) Monitoring
- (9) Automated Testing
- (10) Deployment
- (11) Database

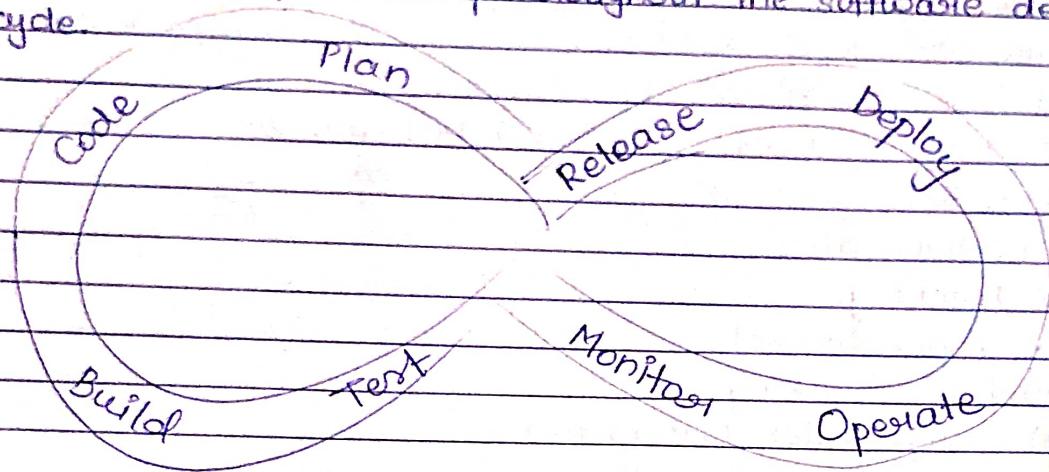
The importance of DevOps ToolChain in software development lies in its ability to

- (1) Improve Collaboration
- (2) Accelerate Delivery
- (3) Enhance Quality

- (a) Increase Agility
 (b) Boost Efficiency

(c) DevOps Architecture

DevOps is a cultural & organizational approach to software development that aims to bridge the gap between development (Dev) & operations team (Ops), thereby improving collaboration, communication, efficiency throughout the software development cycle.



DevOps Architecture is used for the applications hosted on the cloud platform & large distributed applications.

Agile development is used in the DevOps architecture so that integration & delivery can be contiguous.

DevOps enables the team to change their shortcomings & increase productivity.

Below are the components :

Dev

Build

Code

Test

Plan

Ops

Monitor

Deploy

Operate

Release

Features of DevOps Architecture

- (1) Automation
 - (2) Collaboration
 - (3) Integration
 - (4) Configuration Management
- (5) Containerization is a method of packaging, distributing & running applications in a lightweight, isolated environment called as container.

This approach allows applications to run consistently across different environments.

Docker is one of the most popular containerization platforms. It provides tools & a platform to create, deploy & manage containers.

Docker utilizes several key components to implement containerization.

(1) Docker Engine

It is responsible for creating & managing containers on a host system.

It includes a server & a REST API for interacting with the daemon.

(2) Docker Images

Images are read-only templates used to create containers.

(3) Docker Containers

Containers are lightweight, portable & self-sufficient runtime environments created from Docker images.

Each container runs as a separate process, isolated from other containers & the host system.

DATE:

(4) Dockerfile

It is a text file that contains instructions for building a docker image

(5) Docker Registry

Docker Hub is the default public registry where users can find & share docker images

A platform for creating, publishing, distributing, and running containerized applications.

In a containerization technology, Docker is the most popular and widely adopted for creating, publishing, and running containerized applications.

The variety of information and features available in Docker makes it a popular choice for application development and deployment.

Containerization provides several benefits, including portability, scalability, and efficiency.

Portability: Docker containers are designed to run on any host system, making it easy to move them between environments.

Scalability: Docker allows for easy scaling of applications by simply adding more containers to handle increased demand.

Efficiency: Containerization reduces overhead by sharing the underlying infrastructure, which leads to faster boot times and lower resource usage.

Overall, Docker has become a fundamental tool for modern application development and deployment.

It is used in various industries, including finance, healthcare, and e-commerce, to build, test, and deploy complex applications.

With its growing popularity and widespread adoption, Docker continues to play a crucial role in the future of software development.

Overall, Docker is a powerful tool for building, testing, and deploying modern applications.