



Continuous Assessment for Laboratory / Assignment sessions

Academic Year 2022-23

Name: Kaustik Jolapara

SAP ID: 60004200107

Course: Software Engineering

Course Code: **DJ19CEC601**

Year: **T.Y. B.Tech.**

Sem: **VI**

Batch: B1

Department: Computer Engineering

Performance Indicators (Any no. of Indicators) (Maximum 5 marks per indicator)	1	2	3	4	5	6	7	8	9	10	11	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)	4	4	4	4	4	4	4	4	4	4				4	4		
2. Describe (Factual/Conceptual/Procedural/ Metacognitive)	4	-	5	5	4	4	4	5	4	5				4	4		
3. Demonstration (Factual/Conceptual/Procedural/ Metacognitive)	-	-	4	-	-	-	-	-	5	4				4	5		
4. Strategy (Analyse & / or Evaluate) (Factual/Conceptual/ Procedural/Metacognitive)	3	4	-	4	4	3	4	5	-	-	-			3	5		
5. Interpret/ Develop (Factual/Conceptual/ Procedural/Metacognitive)	-	4	4	4	3	4	4	4	-	-				-	-		
6. Attitude towards learning (receiving, attending, responding, valuing, organizing, characterization by value)	4	4	4	4	4	4	-	-	4	5				3	4		
7. Non-verbal communication skills/ Behaviour or Behavioural skills (motor skills, hand-eye coordination, gross body movements, finely coordinated body movements speech behaviours)	4	4	-	-	-	-	4	4	4	5				-	-		
Total	19	20	21	21	19	19	20	22	21	23	-			18	22	40	20
Signature of the faculty member	<i>[Signature]</i>	<i>[Signature]</i>	<i>[Signature]</i>	<i>[Signature]</i>	<i>[Signature]</i>	<i>[Signature]</i>	<i>[Signature]</i>	<i>[Signature]</i>	<i>[Signature]</i>	<i>[Signature]</i>				<i>[Signature]</i>	<i>[Signature]</i>	<i>[Signature]</i>	<i>[Signature]</i>

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

Laboratory marks Σ Avg. = <u>20.5</u>	Assignment marks Σ Avg. = <u>20</u>	Total Term-work (25) = <u>20.5</u>
Laboratory Scaled to (15) = <u>12.5</u>	Assignment Scaled to (10) = <u>8.0</u>	Sign of the Student: <u>Kaustik</u>

Signature of the Faculty member: [Signature]
 Name of the Faculty member:

Signature of Head of the Department
 Date:

Software Engineering

Experiment-1

Div: B

Batch: B1

Team Members:

Meet Patel - 60004200104

Kartik Jolapara - 60004200107

Aim: To identify a suitable life cycle model for your case study and justify your choice

Abstract:

The main aim of this case study is to develop a news application which provides handy news data to the consumers based on their geolocation, making the application interactive and appealing to the consumers while keeping them notified about all the latest news updates. It also gives accurate knowledge about the authenticity of the news and thus predicting whether the news is fake or not.

Features:

1. Providing news to the users based on different categories such as business, sports, entertainment etc. and also bifurcating them as local and global news.
2. Providing news in form of videos of short span duration
3. Local news based on the geolocation of the consumer within certain range and allowing them to subscribe to the particular news
4. Allowing users to post content on the platform so that other users can notify the general public about the recent updates in their locality with a check on nudity, vulgarity and slur content.
5. A recommendation system based on the viewing content of the consumer.

Overview: Vast amount of news data is available every minute throughout the internet. It often happens that the important news heads are missed in this huge lake of data. Hence it becomes quite important to have a platform that can segregate the news which is most relevant to the users depending on the immediate impact it is creating on the user and his/her preferences. Users living in a certain area, based on their location can be directly notified in bulk about news such as train mega blocks, severity of pollution or locality hit health problems. The huge chunk of news can also make the users life easy by having the required checks on the quality of data. This application can also be particularly useful for travel enthusiasts who can get the jist of happenings in their surroundings of which they are totally unaware. All in all it can

make the user experience quite rich with the quality linking of news with the users data.

Process Models : Agile Model, Incremental Model, Spiral model

Our choice: Agile Model

JUSTIFICATION:

Why not Incremental Model:

1. News applications need to be constantly updated with the latest news and information. Incremental development may not be able to keep up with the rapid pace of news delivery and updating.
2. News applications often have a high degree of interdependence between the different components. An incremental development approach may result in frequent disruptions and rework, which can lead to inefficiencies and delays.
3. News applications are expected to be accurate and reliable, and errors or inaccuracies can result in significant consequences. An incremental development approach may increase the risk of errors and inconsistencies, particularly when making changes to the application's core functionality.

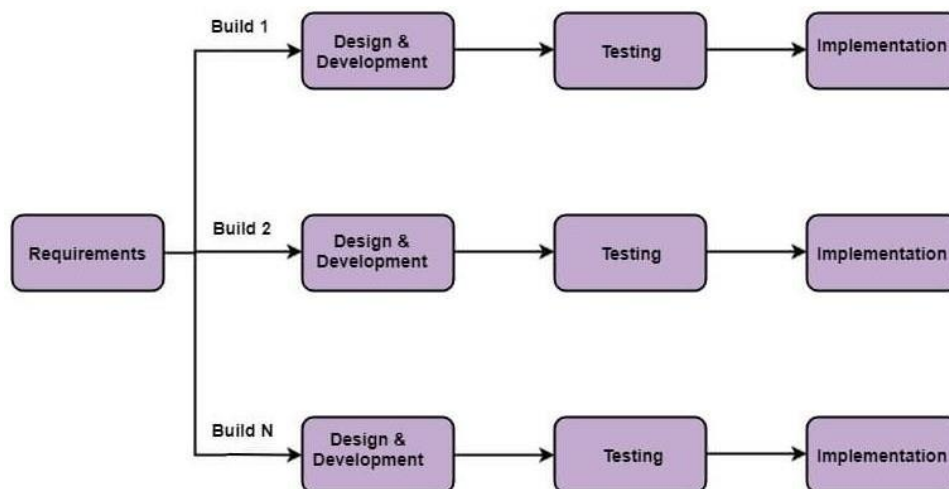
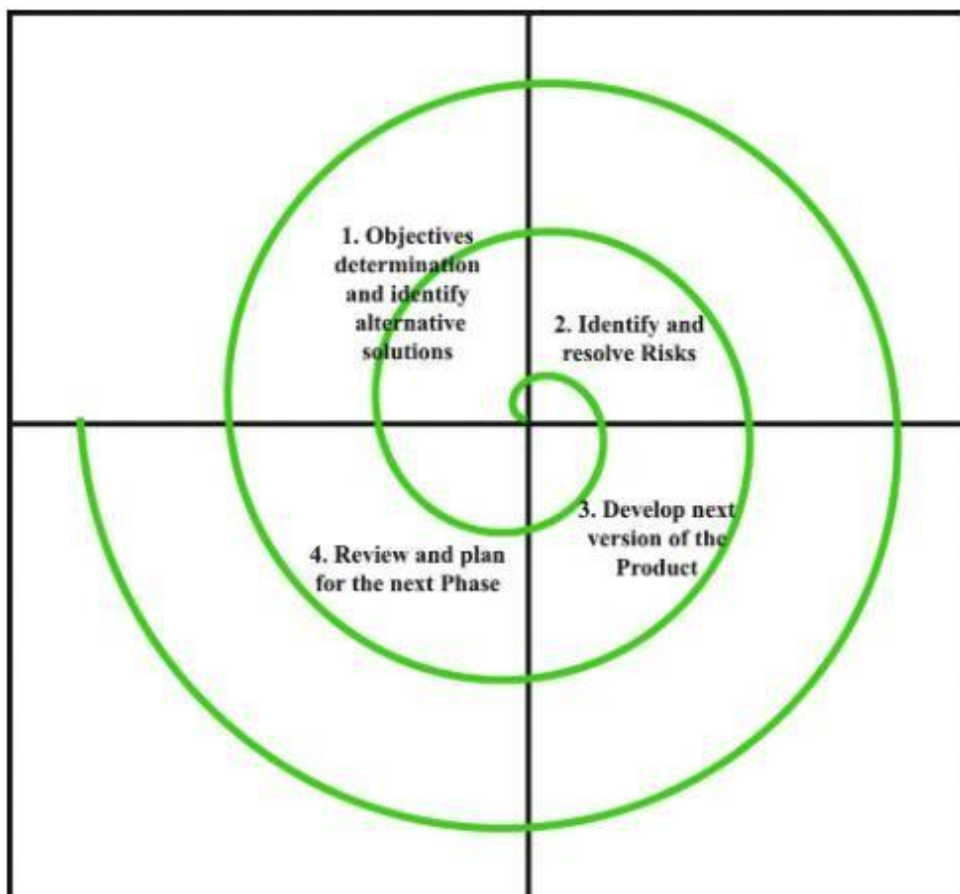


Fig: Incremental Model

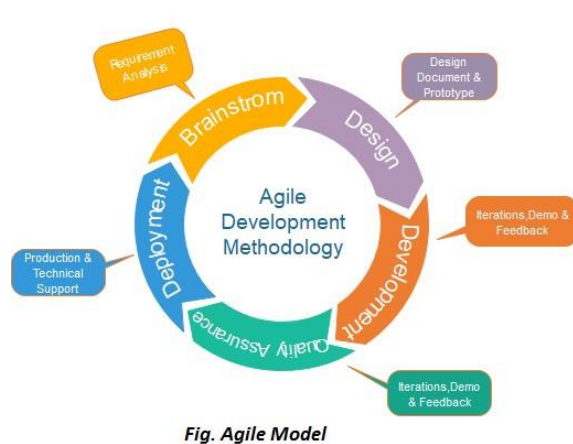
Why not the Spiral Model:

1. News applications need to be developed quickly to keep up with the fast-paced nature of the news industry. The spiral model, with its focus on risk management and iterative development, may not be able to keep up with the required pace of development.
2. News applications have a clear scope and set of requirements that need to be met. The spiral model, with its focus on flexibility and iterative development, may not be well-suited for meeting the specific requirements of a news application.
3. News applications are typically developed on a limited budget, and the spiral model's emphasis on risk management and iterative development may result in increased costs due to additional testing and development cycles.
4. News applications are typically developed on a tight timeline to ensure that they can be released quickly and provide up-to-date information. The spiral model's focus on risk management and iterative development may result in delays that can impact the application's usefulness and relevance.



Why the Agile Model:

1. News applications need to be developed quickly to keep up with the fast-paced nature of the news industry. The Agile model allows for rapid development through iterative cycles of planning, development, testing, and deployment.
2. News applications must be adaptable to changing requirements, and the Agile model's emphasis on flexibility and collaboration allows for changes to be incorporated into the development process quickly.
3. News applications require a collaborative approach to development, involving multiple stakeholders such as journalists, editors, and developers. The Agile model's emphasis on collaboration between cross-functional teams ensures that all stakeholders have a voice in the development process.
4. News applications require continuous improvement to remain relevant, and the Agile model's iterative development approach allows for continuous improvement through regular feedback and testing.
5. News applications must be reliable and accurate to ensure that users can trust the information provided. The Agile model's focus on testing and continuous improvement helps to ensure that the application meets high-quality standards.



Conclusion:

We learnt about the different types of process models and justified the most suitable life cycle model for project execution in our case study.



Software Engineering

Experiment-2

Div: B

Batch: B1

Team Members:

Meet Patel - 60004200104

Kartik Jolapara - 60004200107

Aim: To develop Software Requirement Specification (SRS) document in IEEE format for the project.

Theory:

SRS Software Requirements Specification

A document that specifies most of the requirements as required by the customer and as understood by the software engineer.

A well formatted document that includes scope, purpose, product perspective, software and hardware requirements, functional and non-functional requirements for the product.

Performance:

1. Identify a suitable case study with the scope for software engineering process.
2. Explain the abstract in one page clearly explaining the project with their functionalities.
3. Each project should have atleast 4 functional requirements clearly explaining each functionality by referring to the given SRS template.
4. Prepare a well-formatted document

Conclusion:

In this experiment we were able to make an SRS for our case study and clearly understood the process and requirements for an SRS.

Software Requirements Specification

for

Inshorts

Version 1.0

Prepared by

**Meet Patel
Kartik Jolapara**

**60004200104
60004200107**

**meetp272003@gmail.com
ksjolapara@gmail.com**

Instructor: Dr. Kiran Bhowmick

Course: Software Engineering

Lab Section:

Teaching Assistant:

Date: 12th March 2023

Contents

REVISIONS.....	III
1 INTRODUCTION	5
1.1 DOCUMENT PURPOSE	5
1.2 PRODUCT SCOPE	5
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW	6
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS	6
1.5 DOCUMENT CONVENTIONS.....	6
1.6 REFERENCES AND ACKNOWLEDGMENTS.....	7
2 OVERALL DESCRIPTION.....	7
2.1 PRODUCT PERSPECTIVE.....	7
2.2 PRODUCT FUNCTIONALITY.....	7
2.3 USERS AND CHARACTERISTICS	8
2.4 OPERATING ENVIRONMENT	8
2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS	8
2.6 USER DOCUMENTATION.....	8
2.7 ASSUMPTIONS AND DEPENDENCIES	9
3 SPECIFIC REQUIREMENTS.....	9
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	9
3.2 FUNCTIONAL REQUIREMENTS	13
3.3 BEHAVIOUR REQUIREMENTS.....	13
4 OTHER NON-FUNCTIONAL REQUIREMENTS	14
4.1 PERFORMANCE REQUIREMENTS	14
4.2 SAFETY AND SECURITY REQUIREMENTS	14
4.3 SOFTWARE QUALITY ATTRIBUTES	14
5 OTHER REQUIREMENTS	ERROR! BOOKMARK NOT DEFINED.
APPENDIX A – DATA DICTIONARY	15
APPENDIX B - GROUP LOG	16

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

<In this template you will find text bounded by the “<>” symbols. This text appears in italics and is intended to guide you through the template and provide explanations regarding the different sections in this document. There are two types of comments in this document. These comments that are in black are intended specifically for that course. These comments that are in blue are more general and apply to any SRS. Please, make sure to delete all of the comments before submitting the document.

The explanations provided below, do not cover all of the material, but merely, the general nature of the information you would usually find in SRS documents. It is based on the IEEE requirements and was adapted specifically for the needs of Software Engineering 3K04/3M04 courses. Most of the sections in this template are required sections, i.e. you must include them in your version of the document. Failure to do so will result in marks deductions. Optional sections will be explicitly marked as optional.



Academic Year: 2021_22

1 Introduction

The main aim of this project is to develop a news application which provides handy news data to the consumers based on their geolocation, making the application interactive and appealing to the consumers while keeping them notified about all the latest news updates. It also gives accurate knowledge about the authenticity of the news and thus predicting whether the news is fake or not. Vast amount of news data is available every minute throughout the internet. It often happens that the important news heads are missed in this huge lake of data. Hence it becomes quite important to have a platform that can segregate the news which is most relevant to the users depending on the immediate impact it is creating on the user and his/her preferences. Users living in a certain area, based on their location can be directly notified in bulk about news such as train mega blocks, severity of pollution or locality hit health problems. The huge chunk of news can also make the users life easy by having the required checks on the quality of data. This application can also be particularly useful for travel enthusiasts who can get the gist of happenings in their surroundings of which they are totally unaware. All in all it can make the user experience quite rich with the quality linking of news with the users data

1.1 Document Purpose

This is a SRS for the project Inshorts- Version 1.0. The purpose of this Software Requirement Specifications document is to clearly define the technical aspect of the product we intend to build. It covers how the first version of the application will interact with the users, hardware, software and other interconnecting applications. It specifies the requirements, technical details and the limitations of the project which shall help us ensure we utilize all the factors correctly, satisfy all product requirements and provide the best possible product to our users.

1.2 Product Scope

A news application is a software product designed to provide users with the latest news and information from various sources. The scope of a news application can include features such as user profiles, a news feed, push notifications, search functionality, bookmarking, sharing, commenting, settings, analytics, and monetization. Users can create profiles on the news application to customize their news feed according to their interests, preferences, and location. The news feed displays the latest news stories from various sources, and users can filter their news feed based on categories such as politics, sports, entertainment, business, and more. Push notifications are sent to users for breaking news stories or important events in their selected categories or from their preferred news sources. Users can also search for specific news stories or topics, bookmark their favorite news stories, and share them with others. The news application can also include features such as comments,



Academic Year: 2021_22

settings, and analytics to track user engagement and performance metrics. Finally, the news application can generate revenue through various monetization models, such as advertising, subscriptions, or in-app purchases.

1.3 Intended Audience and Document Overview

The document is organised in a climactic sequence. The document opens with a brief abstract that introduces the theme and sets the tone for the subsequent parts. Following that, the document describes its characteristics and introduces the issue statement. It then goes on to provide an overview of the project's users, environment, and limits. The document then goes on to describe the different interfaces and project needs. The document also includes references to pertinent research publications and sources. The document is intended for all users as well as the project's technical developers. The primary audience for documentation of a news application is typically the developers, testers, and other technical personnel who are responsible for building, maintaining, and supporting the application. This audience includes developers who are responsible for designing and building the news application, testers who ensure that the application works correctly and meets requirements, technical writers who create documentation, and support staff who provide technical assistance to users. Managers who oversee the development and maintenance of the news application may also need to understand how it works to make informed decisions about its future direction. Overall, the documentation should be clear and comprehensive, providing technical personnel with information on how the application works, how to install and configure it, and how to use its features effectively. By providing clear documentation, the technical personnel can ensure that the news application is built, tested, and supported correctly, leading to a successful and reliable application.

1.4 Definitions, Acronyms and Abbreviations

Some of the terms that can be frequently encountered in the SRS are listed below:

- **API** – Application Programming Interface
- **GCP** – Google Cloud Platform
- **GPS** – Global Positioning System
- **Behavior Analysis** - Finding recurring patterns in data for future recommendations.
- **GUI** – Graphical User Interface

1.5 Document Conventions

The following conventions were followed while creating the document:

- We have used the IEEE standards for document formatting.



Academic Year: 2021_22

- The font used is Arial, font size for title is 14 and font size for text is 12.
- Italics have been used for comments.
- 1" margin has been maintained throughout the document.
- The text is single spaced.

1.6 References and Acknowledgments

These are some of the references:

<https://developer.apple.com/design/human-interface-guidelines/news/>
<https://material.io/design/guidelines-for-news/>
<https://reutersinstitute.politics.ox.ac.uk/digital-news-report>
<https://www.pewresearch.org/topics/journalism-and-media/>
<https://www.americanpressinstitute.org/>
<https://www.internationalnewsservices.com/>
<https://journalists.org/>

2 Overall Description

2.1 Product Perspective

The app offers a personalized experience that caters to the user's preferences, allowing them to select topics of interest, save articles, and receive notifications for breaking news. The content is curated to provide high-quality, relevant news from trusted sources, with options for users to explore different perspectives and formats. Monetization strategies includes advertising, subscriptions, and is designed to balance user satisfaction with business goals. Analytics is used to measure engagement and identify areas for improvement, with regular updates and feature enhancements to keep the app fresh and compelling.

2.2 Product Functionality

- News Reels: Users can watch news reels to stay updated with the latest news.
- Personalization: Users can select their preferences and receive news tailored to their interests.
- Location-Based News: Users can receive news based on their location to stay informed about local news and events.
- User-Generated Content: Users can add news to the app, providing a platform for citizen journalism and community engagement.
- Breaking News Notifications: Users can receive notifications for breaking news stories, keeping them informed in real-time.
- Bookmarking and Saving: Users can bookmark and save articles for future reference, making it easy to come back and read articles at a later time.



Academic Year: 2021_22

- **Search:** Users can search for specific news stories, topics, or keywords to quickly find the information they need.
- **Multilingual Support:** The app can support multiple languages, making it accessible to a broader audience.
- **Social Sharing:** Users can share news articles on social media platforms, increasing the app's reach and engagement.

2.3 Users and Characteristics

- **Users:** They are the local users who want to be updated with the current local and global news. This includes all the Sports fans, Business professionals, Social media users, Local news enthusiasts, News junkies, Information seekers, Casual readers.

2.4 Operating Environment

- **Recommended browsers:** Chrome, Firefox, Safari, Edge and Brave.
- **Recommended Operating systems:** Windows, MacOS, IpadOS, iOS, wear OS, watchOS, Android and Linux.

2.5 Design and Implementation Constraints

Design and implementation constraints are essential factors that need to be considered during the development of a news application. The design constraints of a news application require it to have an intuitive and user-friendly interface that enables users to navigate and consume news stories seamlessly. Also, the application should be responsive and fast-loading, allowing for a smooth user experience. Customization is also an important design constraint to consider, as users may prefer different types of news, sources, and topics. Additionally, the application should be accessible to users with disabilities to ensure inclusivity.

Application constraints are also crucial considerations for a news application. The application may need to store large amounts of data, including news stories and user preferences, which can impact its performance and scalability. Security is another critical application constraint to consider, as the application should be designed to protect user data and prevent unauthorized access or data breaches. The application may also need to integrate with third-party services, which can introduce additional complexity and constraints. Performance is also an important application constraint, as the application needs to be optimized for handling large volumes of traffic and delivering news stories quickly and reliably.

2.6 User Documentation

- The user manual will contain all the guidelines for handling software as well as FAQ section for reference.



Academic Year: 2021_22

- Contact us & support center.

2.7 Assumptions and Dependencies

- User Preferences: If the assumed user preferences are incorrect, users may not find the content relevant or interesting, leading to lower engagement and retention.
- Content Quality: If the assumed quality of the news content does not meet the user's expectations, it can lead to lower engagement, lower retention, and negative reviews.
- Monetization Strategy: If the assumed monetization strategy is not effective, it can lead to lower revenue, user dissatisfaction, and the need to pivot the business model.
- Technical Requirements: If the assumed technical requirements for the app are wrong, it can lead to performance issues, user frustration, and negative reviews.
- Regulatory Compliance: If the assumed regulatory requirements are incorrect, it can lead to legal and financial repercussions, which can negatively impact the app's reputation and revenue.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

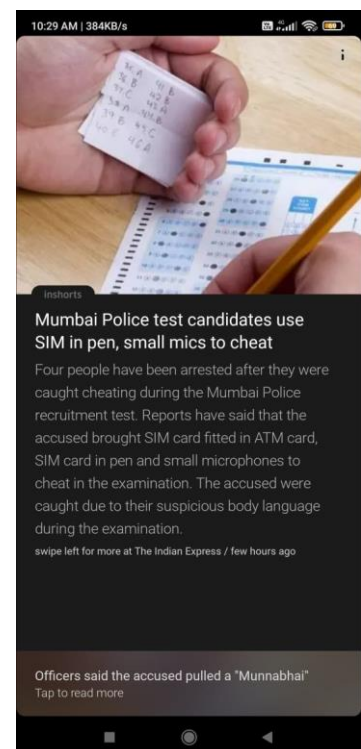
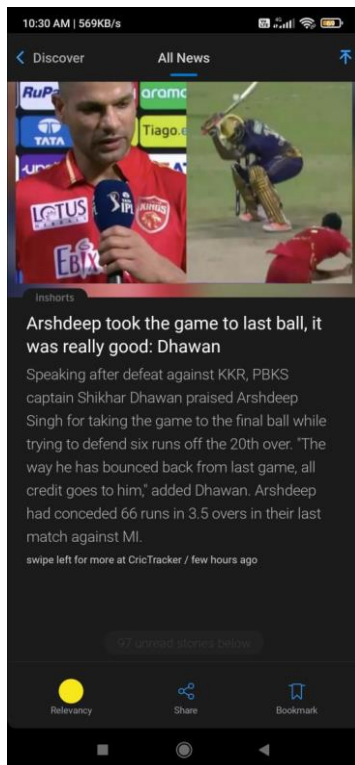
Mobile App Screenshots



**SHRI VILEPARLE 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)



Academic Year: 2021_22



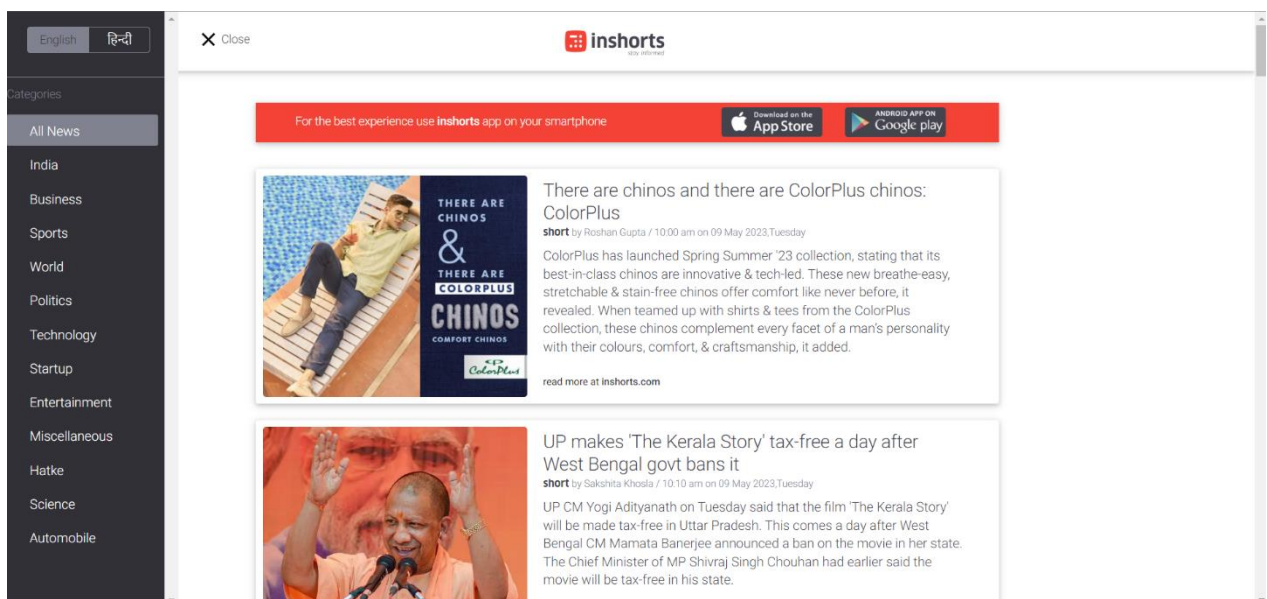
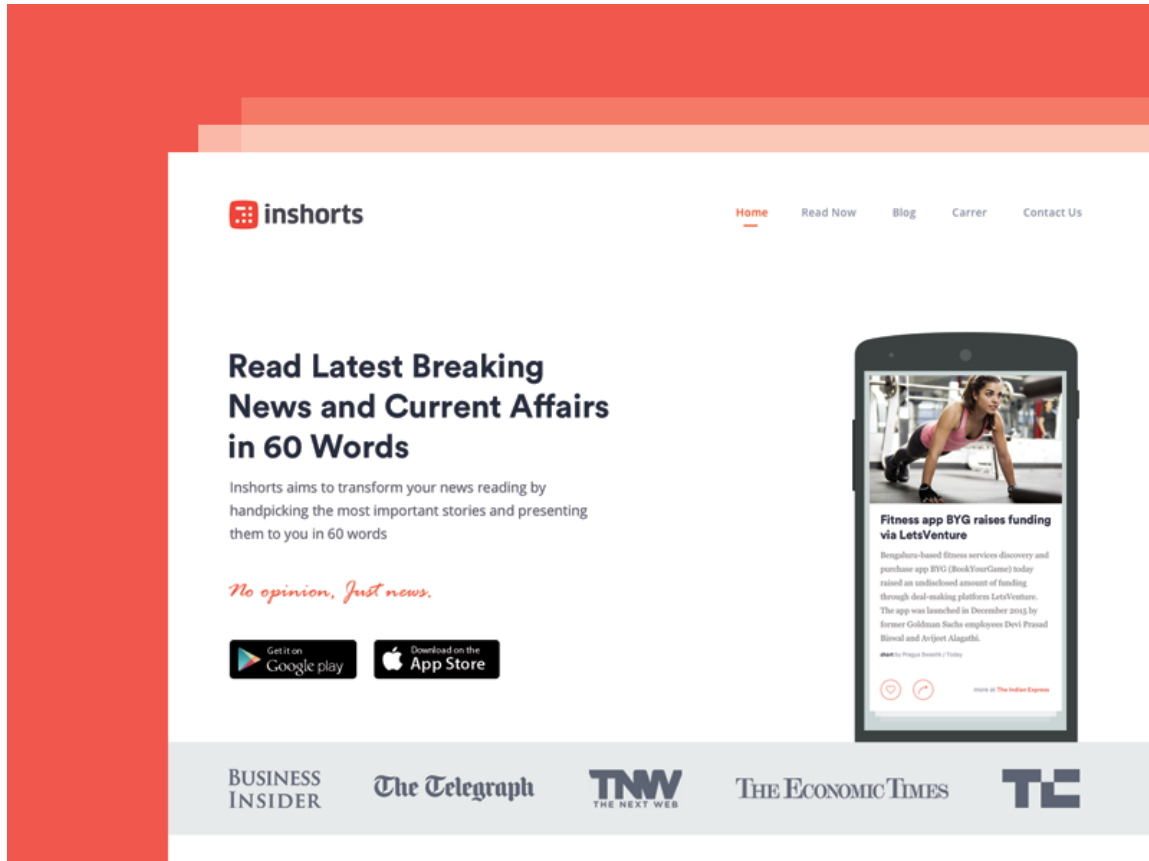


SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)



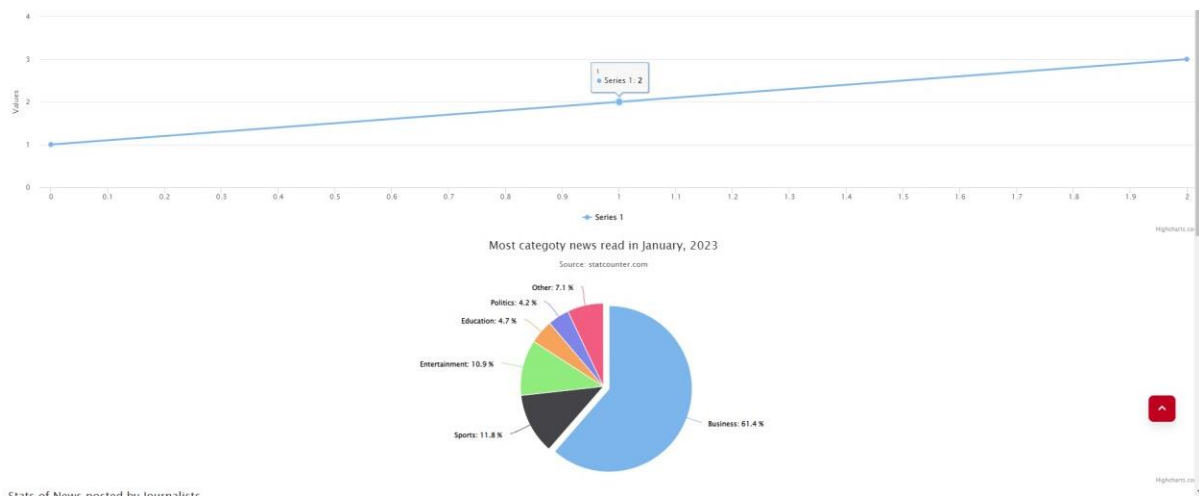
Academic Year: 2021_22

Website Screenshots:





Academic Year: 2021_22



3.1.2 Hardware Interfaces

Server:

RAM: 8GB

Storage : 1 TB SSD

Processor: Intel Pentium 4 processor or later that's SSE2 capable

GPU: Nvidia GTX 1050

User Device:

RAM: 6 GB

Storage: 128 GB Storage

GPS Sensor

3.1.3 Software Interfaces

Browsers: Chrome, Firefox, Safari, Edge and Brave.

Operating systems: Windows, MacOS, IpadOS, iOS, wear OS, watchOS, Android and Linux.

Tools: Google Colab, Jupyter Notebook

3.1.4 Communications Interfaces

- Minimum 40 Kbps Internet Speed to ensure lossless connectivity.
- HTTP protocols for servicing the requests and for transmission of data in JSON format.
- AES protocol will be used to encrypt the sensitive data being transmitted.



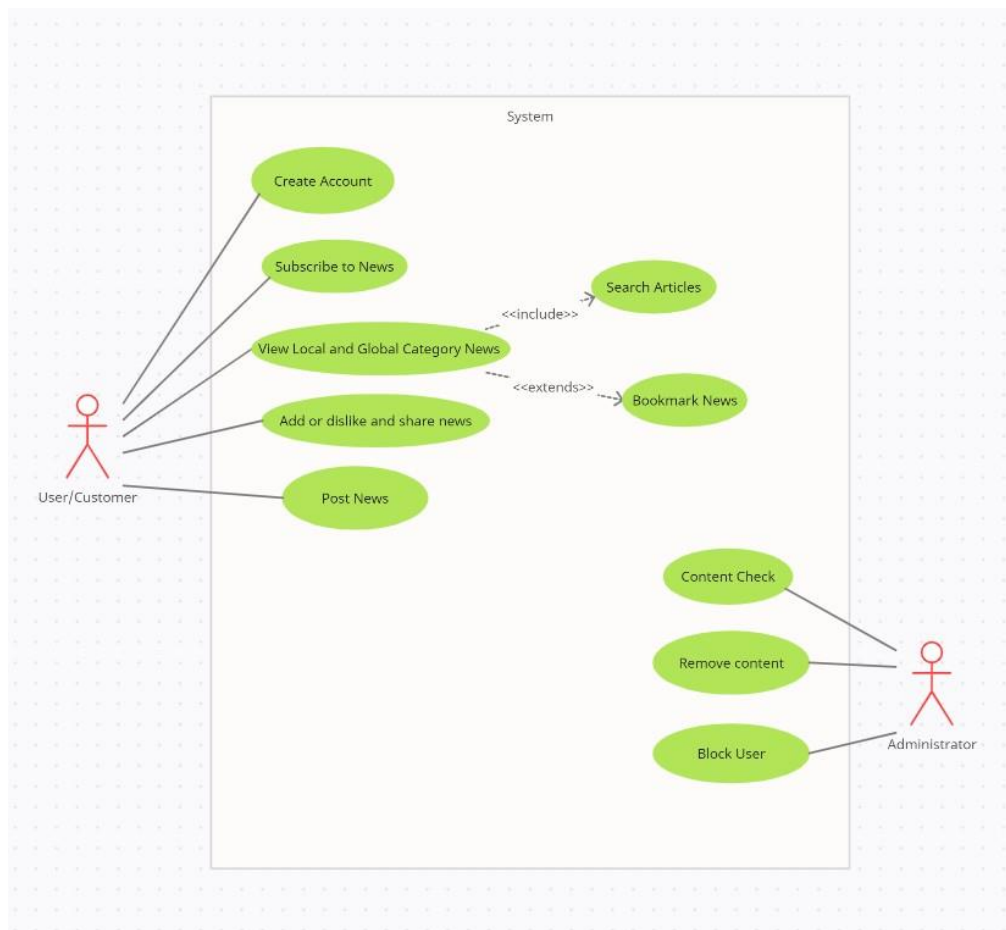
Academic Year: 2021_22

3.2 Functional Requirements

- **Login / Signup** : This functionality will be used by users to register as well as to login in the platform. All the sensitive credentials will be transmitted and stored in a secure manner.
- **Data Collection** : Geotagged images from social media channels will be collected and stored along with popular nearby news posts and will be deployed of Google Cloud Platform.
- **Bulletin Updation** : Local Authorities will be able to share live updates and details about spots.
- **Recommendation** : Based on his news viewing history, locations will be recommended to the users along with the news category in which user is interested in.

3.3 Behaviour Requirements

3.3.1 Use Case View





4 Other Non-functional Requirements

4.1 Performance Requirements

- The analysis should be crystal clear.
- It should provide the most suitable news available when required.
- It should be able to extract data from users at frequent intervals and suggest them other news according to their liking.
- It should check the news posted by users by having a check at vulgar and inappropriate content.
- It should have support for an ample amount of concurrent users.
- Any bugs/queries should be resolved in 3 days.

4.2 Safety and Security Requirements

- 1) Authentication and Authorization: Implement a strong authentication system that requires users to create unique login credentials, including usernames and passwords. Also, ensure that users can only access the information that is authorized for their account.
- 2) Secure APIs: Implement secure APIs to ensure that third-party applications do not have unauthorized access to user data. APIs should require authentication and should only allow authorized applications to access specific data.
- 3) Regular Updates: Keep the application updated with the latest security patches and upgrades to address any potential security vulnerabilities.
- 4) Security Testing: Perform regular security testing and assessments to identify and address any security vulnerabilities before they can be exploited by attackers.
- 5) User Education: Educate users about safe browsing habits and encourage them to use strong passwords and two-factor authentication to protect their accounts.

4.3 Software Quality Attributes

- **RELIABLE** : Should have sufficient accuracy that the users can rely on it. It should meet client satisfaction standards and be able to gain and maintain their trust.
- **AVAILABILITY** : Whenever the need is there for the analysis it should be available. It should not provide misleading information when it is required the most and function seamlessly.
- **SECURE** : It should have security to ensure it is not tampered with and is not used for illegal purposes.
- **MAINTAINABILITY** : The analysis should be easily maintainable, the users should be able to retrieve and add their news stories and should be able to update them with ease at a later time.



Academic Year: 2021_22

Appendix A – Data Dictionary

Field Name	Data Type	Field Size for display	Description	Example
article_id	Integer	5	unique identifier for the article	543
title	String	10	title of the article	New Study Shows Benefits of Exercise
description	String	50	brief description of the article	A new study published in the Journal of Health and Fitness reveals that regular exercise can have significant benefits on both physical and mental health.
content	String	500	full text of the article	According to the study, individuals who engaged in moderate to vigorous exercise for at least 30 minutes a day showed...
author	String	20	name of the article's author	Jane Doe
date_published	Date	10	date the article was published	2022-02-01
category	List	25	category of the article	[Health , Fitness]
user_id	Integer	5	unique identifier for the user who favorited the article	9875



Academic Year: 2021_22

Appendix B - Group Log

Date	Actors	Work Done
02/03/2023	Meet, Kartik	Analysed Requirements
12/03/2023	Meet, Kartik	Prepared SRS



Software Engineering

Experiment-3

Div: B

Batch: B1

Team Members:

Meet Patel - 60004200104

Kartik Jolapara - 60004200107

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

Theory:

Use case Diagram

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behaviour (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (i.e., use case diagram). A key concept of use case modelling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behaviour in the user's terms by specifying all externally visible system behaviour. Use cases represent only the functional requirements of a system. Other requirements such as business rules, quality of service requirements, and implementation constraints must be represented separately, again, with other UML diagrams.

Use case diagrams are typically developed in the early stage of development and people often apply use case modelling for the following purposes:

- Specify the context of a system.
- Capture the requirements of a system.
- Validate a system architecture.
- Drive implementation and generate test cases.
- Developed by analysts together with domain experts.



Class Diagram

In software engineering, a class diagram in the Unified Modelling Language (**UML**) is **a type of static structure diagram** that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

Purpose of Class Diagrams

1. Shows static structure of classifiers in a system.
2. Diagram provides a basic notation for other structure diagrams prescribed by UML.
3. Helpful for developers and other team members too
4. Business Analysts can use class diagrams to model systems from a business perspective.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.



FOR OUR PROJECT:

Use Case Diagram:

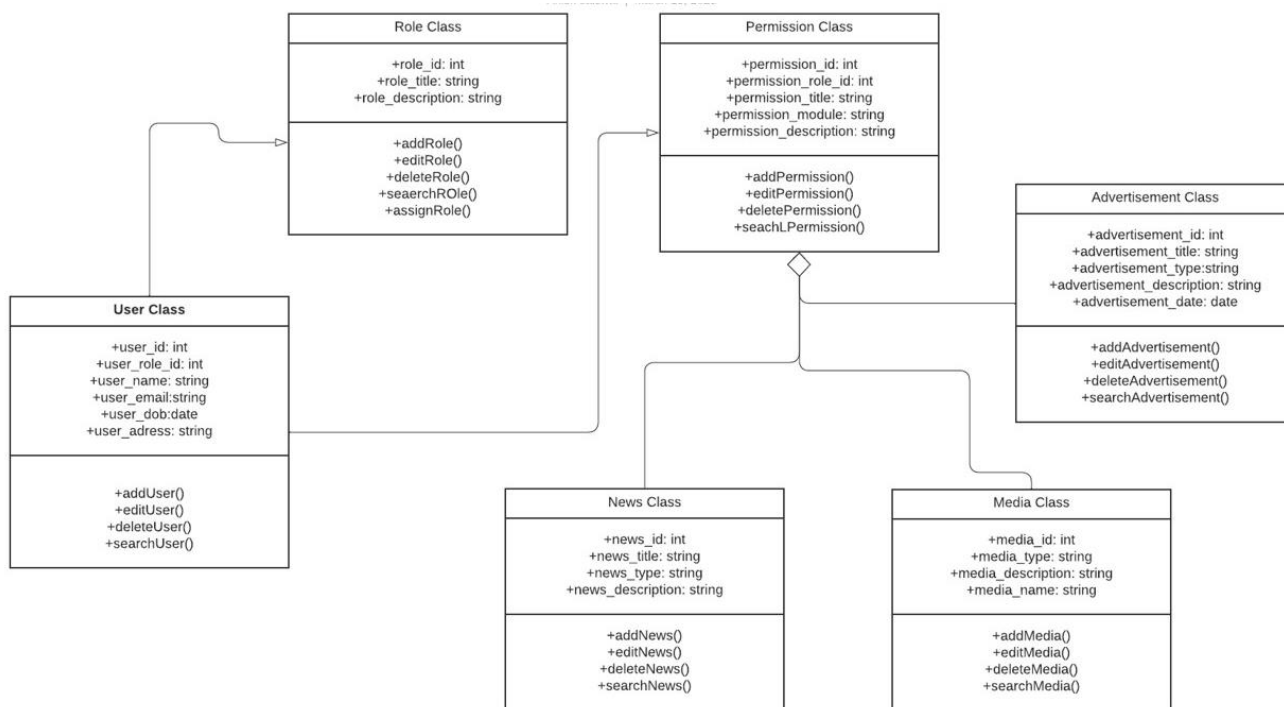


Actors and their User scenarios:

- **Users** will have access to the deployed software and can enter their location to get the desired useful news. Users will also have the privilege to add/subscribe to a particular news and to post news.
- **Administrator** will keep a close eye on the type of the news posted and will have the total rights to remove content of certain type or block the user if needed.
- **Database Manager** will have several use cases:
 - a. Having authentication check: It becomes useful for the database administrator to provide permittable access to the verified users only.
 - b. Managing Users Post data: Saving the data created by the User so as to make it available for all the users. This will also include keeping the count of likes and dislikes on each of the users post.
 - c. He will also perform analysis on the users data so as to suggest him similar trends in the future.
 - d. He will also be handling the recovery and backup of the entire database.



Class Diagram:



- **Class - User**

Attributes:

- User_id
- User_role_id
- User_name
- User_email
- User_dob
- User_address

Operations:

- addUser()
- editUser()



- deleteUser()
- searchUser()

- **Class- Role Class**

Attributes:

- Role_id
- Role_title
- Role_description

Operations:

- addRole()
- editRole()
- deleteRole()
- searchRole()
- assignRole()

- **Class – Permission Class**

Attributes:

- permission_id
- permission_role_id
- permission_title
- permission_module
- permission_description

Operations:

- addPermission()
- editPermission()
- deletePermission()
- searchPermission()



- **Class – Advertisement class**

Attributes:

- advertisement_id
- advertisement_title
- advertisement_type
- advertisement_description
- advertisement_date

Operations:

- addAdvertisement()
- editAdvertisement ()
- deleteAdvertisement ()
- searchAdvertisement ()

- **Class – Media Class**

Attributes:

- media_id
- media_type
- media_description
- media_name

Operations:

- addMedia()
- editMedia()
- deleteMedia ()
- searchMedia()

- **Class – News Class**

Attributes:

- News_id
- News_title
- News_type
- News_description

Operations:

- addNews()
- editNews()
- deleteNews ()
- searchNews()



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)



Conclusion: We learnt about UML Use-Case Diagrams and Class Diagrams and have created them for our tourism analysis project.



Software Engineering

Experiment-4

Div: B

Batch: B1

Team Members:

Meet Patel - 60004200104

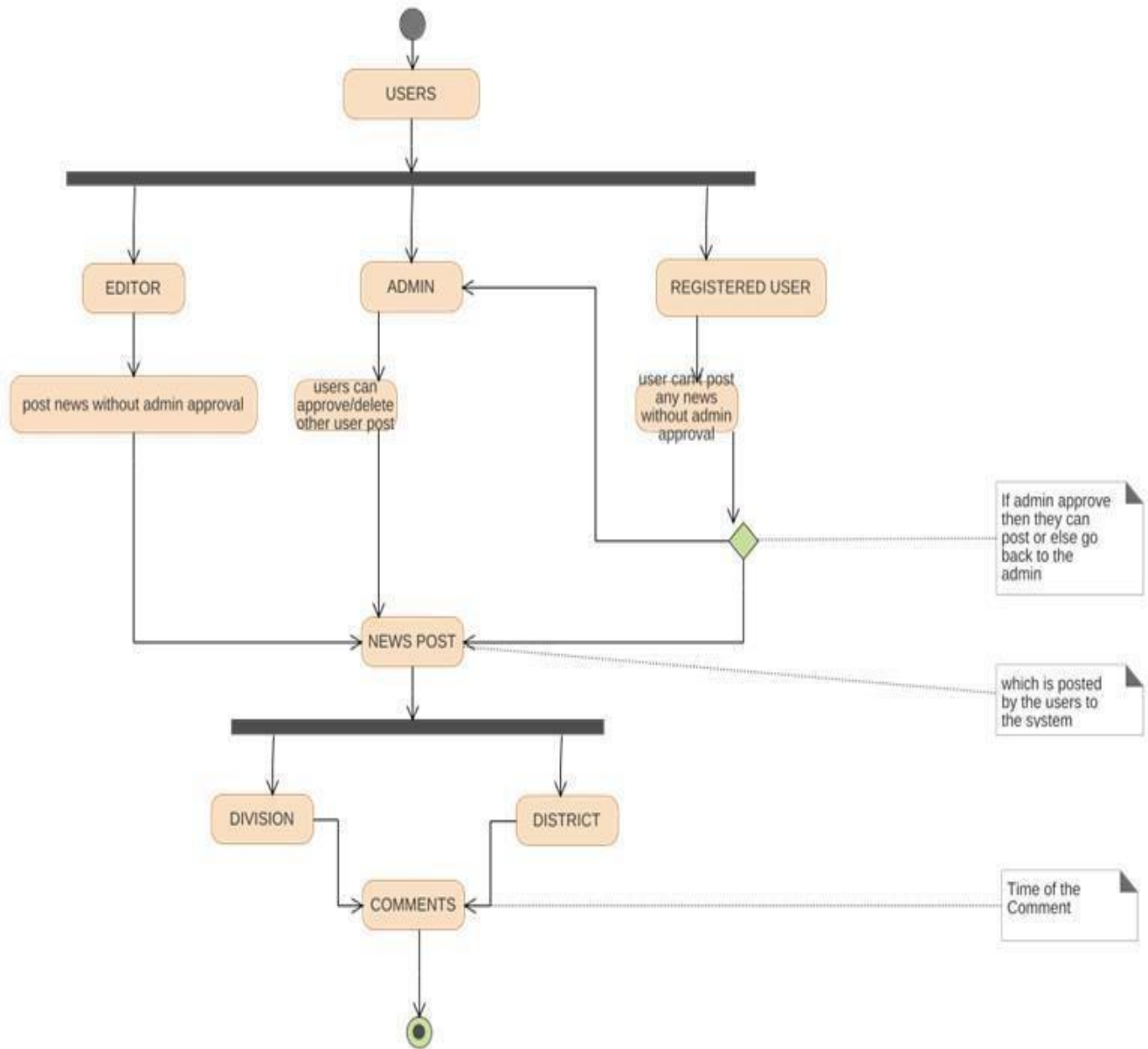
Kartik Jolapara - 60004200107

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

THEORY:

Activity Diagram:

A UML activity diagram depicts the dynamic behavior of a system or part of a system through the flow of control between actions that the system performs. It is similar to a flowchart except that an activity diagram can show concurrent flows. The main component of an activity diagram is an action node, represented by a rounded rectangle, which corresponds to a task performed by the software system. Arrows from one action node to another indicate the flow of control. That is, an arrow between two action nodes means that after the first action is complete the second action begins. A solid black dot forms the initial node that indicates the starting point of the activity. A black dot surrounded by a black circle is the final node indicating the end of the activity. A fork represents the separation of activities into two or more concurrent activities. It is drawn as a horizontal black bar with one arrow pointing to it and two or more arrows pointing out from it. Each outgoing arrow represents a flow of control that can be executed concurrently with the flows corresponding to the other outgoing arrows. These concurrent activities can be performed on a computer using different threads or even using different computers.

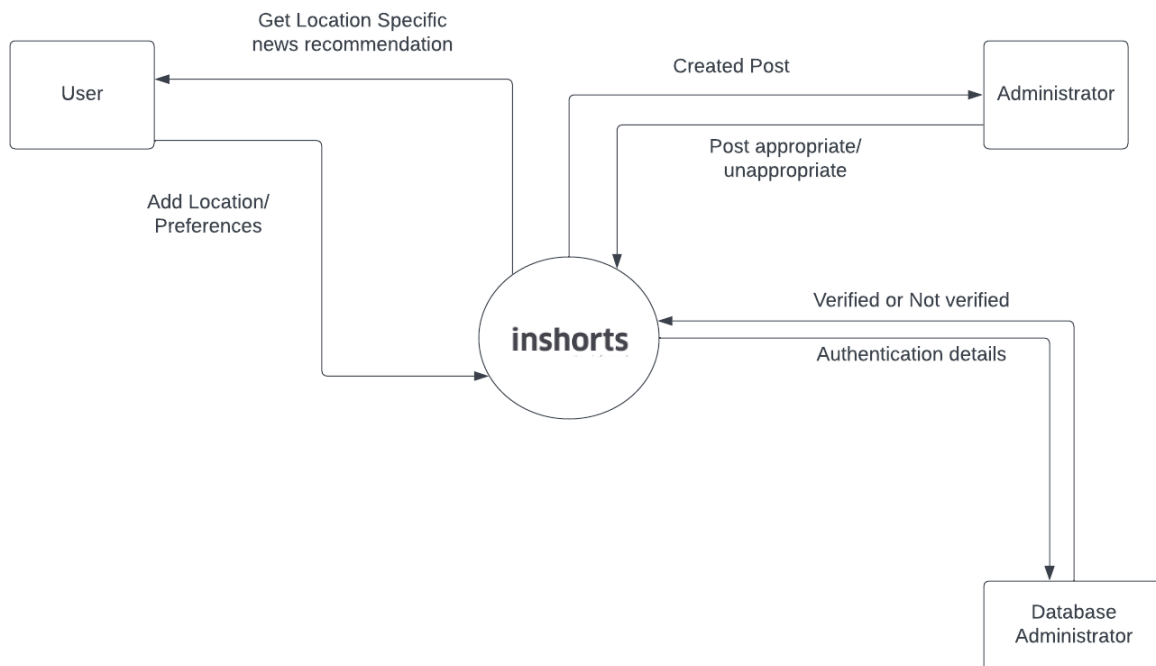




Data Flow Diagrams:

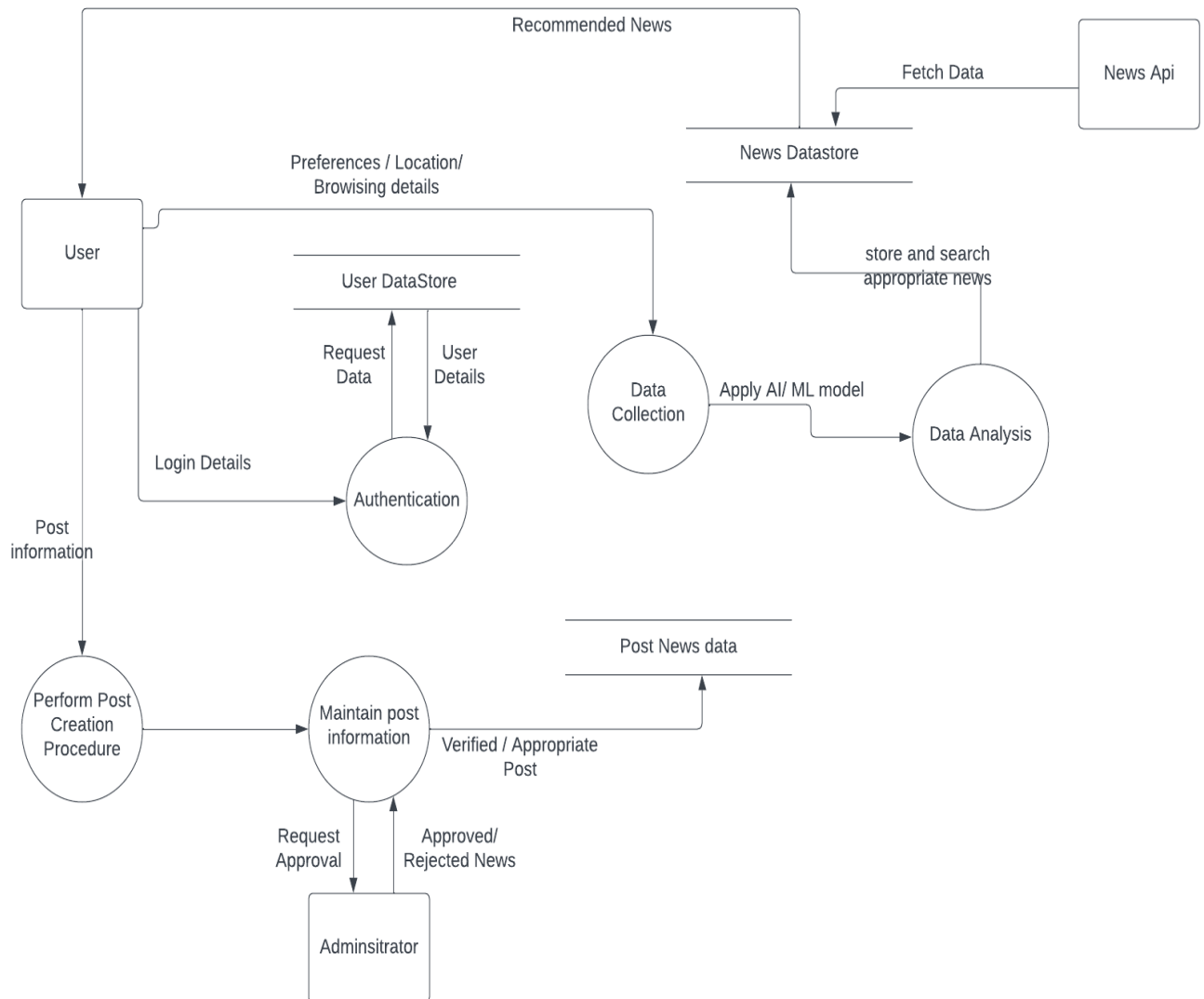
The data flow diagram enables you to develop models of the information domain and functional domain. As the DFD is refined into greater levels of detail, you perform an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of data as it moves through the processes that embody the application.

DFD Level 0 Diagram

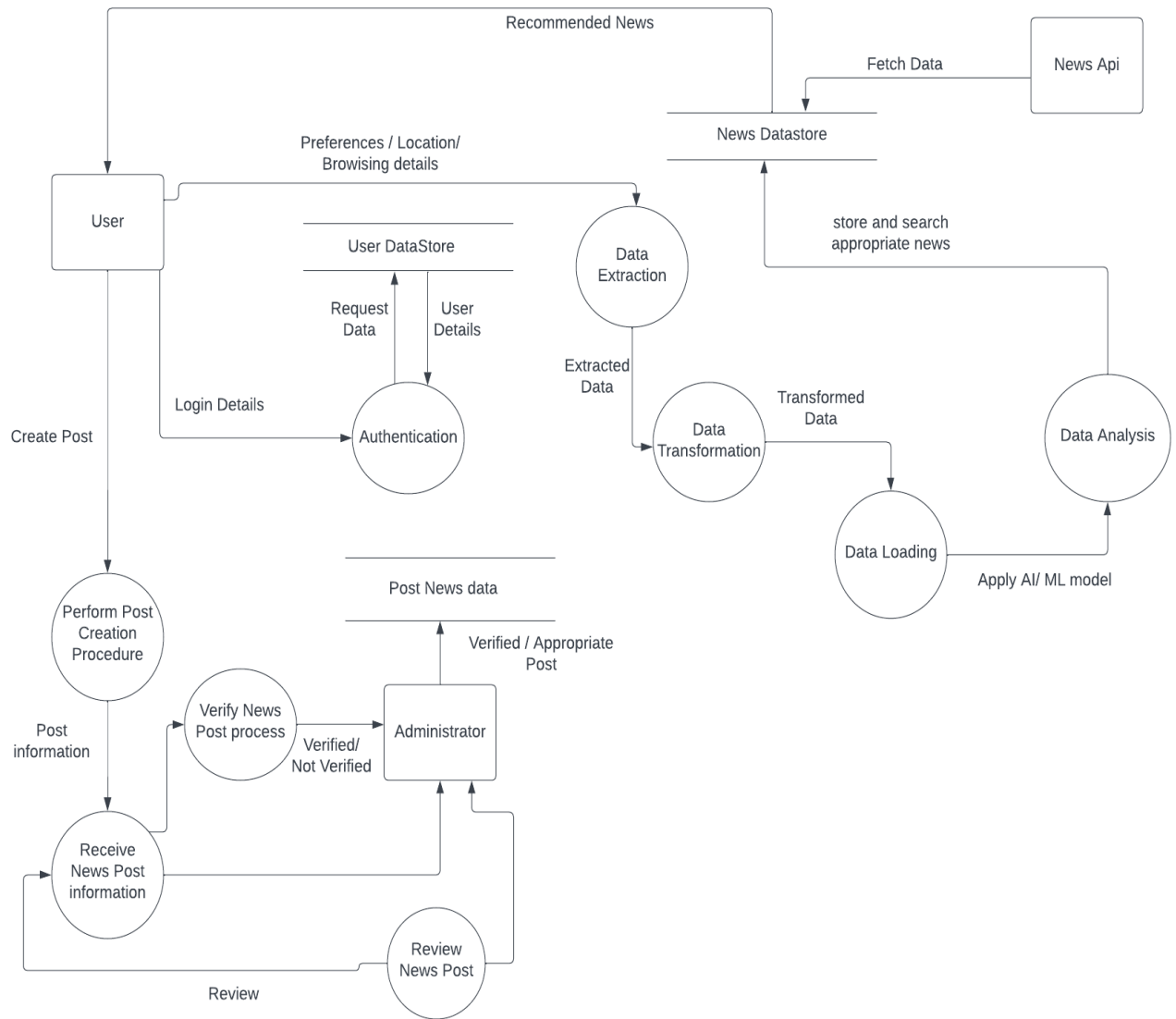




DFD Level 1 Diagram



DFD Level 2 Diagram



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.



Academic Year: 2021_22

Software Engineering

Experiment-5

Div: B

Batch: B1

Team Members:

Meet Patel - 60004200104

Kartik Jolapara - 60004200107

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

Theory:

Work Breakdown Structure:

Work Breakdown Statement

A work breakdown statement (WBS) is a categorized list of tasks with an estimate of resources required to complete the task. An example WBS appears below.

WBS #	Task Description	Est Person (in hrs)	Who	Resources	Materials & Supplies
1	Extract Data from NewsApi	100	Researcher	Extraction Tools	\$150
2	Analyze the extracted data and validate	80	Analyst	Excel, Tableau	\$100
3	Update the data for any inaccuracies	50	Local Authorities	Excel, Tableau	\$100
4	Testing the functionalities	150	Marketing Manager and Analyst	Testing Tools	\$200
5	Project Deployment	72	Marketing Manager and Analyst	Servers	\$500
6	Maintain data pipelines	4000	Backend Engineer	Servers	\$1000

Gantt Chart Basics

Gantt charts are a project planning tool that can be used to represent the timing of tasks required to complete a project. Because Gantt charts are simple to understand and easy to construct, they are used by most project managers for all but the most complex projects.



Academic Year: 2021_22

In a Gantt chart, each task takes up one row. Dates run along the top in increments of days, weeks or months, depending on the total length of the project. The expected time for each task is represented by a horizontal bar whose left end marks the expected beginning of the task and whose right end marks the expected completion date. Tasks may run sequentially, in parallel or overlapping.

As the project progresses, the chart is updated by filling in the bars to a length proportional to the fraction of work that has been accomplished on the task. This way, one can get a quick reading of project progress by drawing a vertical line through the chart at the current date. Completed tasks lie to the left of the line and are completely filled in. Current tasks cross the line and are behind schedule if their filled-in section is to the left of the line and ahead of schedule if the filled-in section stops to the right of the line. Future tasks lie completely to the right of the line.

In constructing a Gantt chart, keep the tasks to a manageable number (no more than 15 or 20) so that the chart fits on a single page. More complex projects may require subordinate charts which detail the timing of all the subtasks which make up one of the main tasks. For team projects, it often helps to have an additional column containing numbers or initials which identify who on the team is responsible for the task.

Often the project has important events which you would like to appear on the project timeline, but which are not tasks. For example, you may wish to highlight when a prototype is complete or the date of a design review. You enter these on a Gantt chart as "milestone" events and mark them with a special symbol, often an upside-down triangle.

Gantt Chart Example from <http://www.criticaltools.com/projwbs.htm>

Practical:

For Estimation

1. Use the FP / COCOMO model to estimate Effort and subsequently Cost required to develop the project.
2. Show all the tables and steps of the estimation model.

Cost estimation

- E.I.: User credentials, User news posts, User comments, Admin credentials
- E.O.: Searched news, Location based news, Recommended news
- E.Q.: News search



Academic Year: 2021_22

- I.L.F.: User data, News data, Location data, Recommender system
- E.I.F: -

Information Domain Value	Count	Simple	Average	Complex	Total
External inputs	4	3	<u>4</u>	8	4*4=16
External Outputs	3	4	<u>6</u>	9	6*3=18
External enquiry	1	2	<u>3</u>	6	3*1=3
Internal Logical Files	4	3	<u>5</u>	9	7*4=20
External interface files	0	4	<u>7</u>	10	7*0=0
					Count-total = 57

The F_i ($i = 1$ to 14) are value adjustment factors (VAF) based on responses to the following questions:

1. Does the system require reliable backup and recovery?

- 4 - It is essential to have reliable backup and recovery for important modules.

2. Are specialized data communications required to transfer information to or from the application?

- 5 – Application needs to be “realtime” .

3. Are there distributed processing functions?

- 3 - Yes

4. Is performance critical?

- 4 - Performance is always critical



Academic Year: 2021_22

5. Will the system run in an existing, heavily utilized operational environment?
 - 5 - No, the project is not based on a pre-existing system.
6. Does the system require online data entry?
 - 4 - Yes
7. Does the online data entry require the input transaction to be built over multiple screens or operations?
 - 3 - Yes, search of news by users will be built over multiple screens.
8. Are the ILFs updated online?
 - 5
9. Are the inputs, outputs, files, or inquiries complex?
 - 3 - Yes
10. Is the internal processing complex?
 - 4 - Yes
11. Is the code designed to be reusable?
 - 4 - Yes as we are following the best coding practices.
12. Are conversion and installation included in the design?
 - 3
13. Is the system designed for multiple installations in different organizations?
 - 0 - No
14. Is the application designed to facilitate change and ease of use by the user
 - 4 - Yes

$\Sigma(F_i) = 51$

The estimated number of FP is derived:



Academic Year: 2021_22

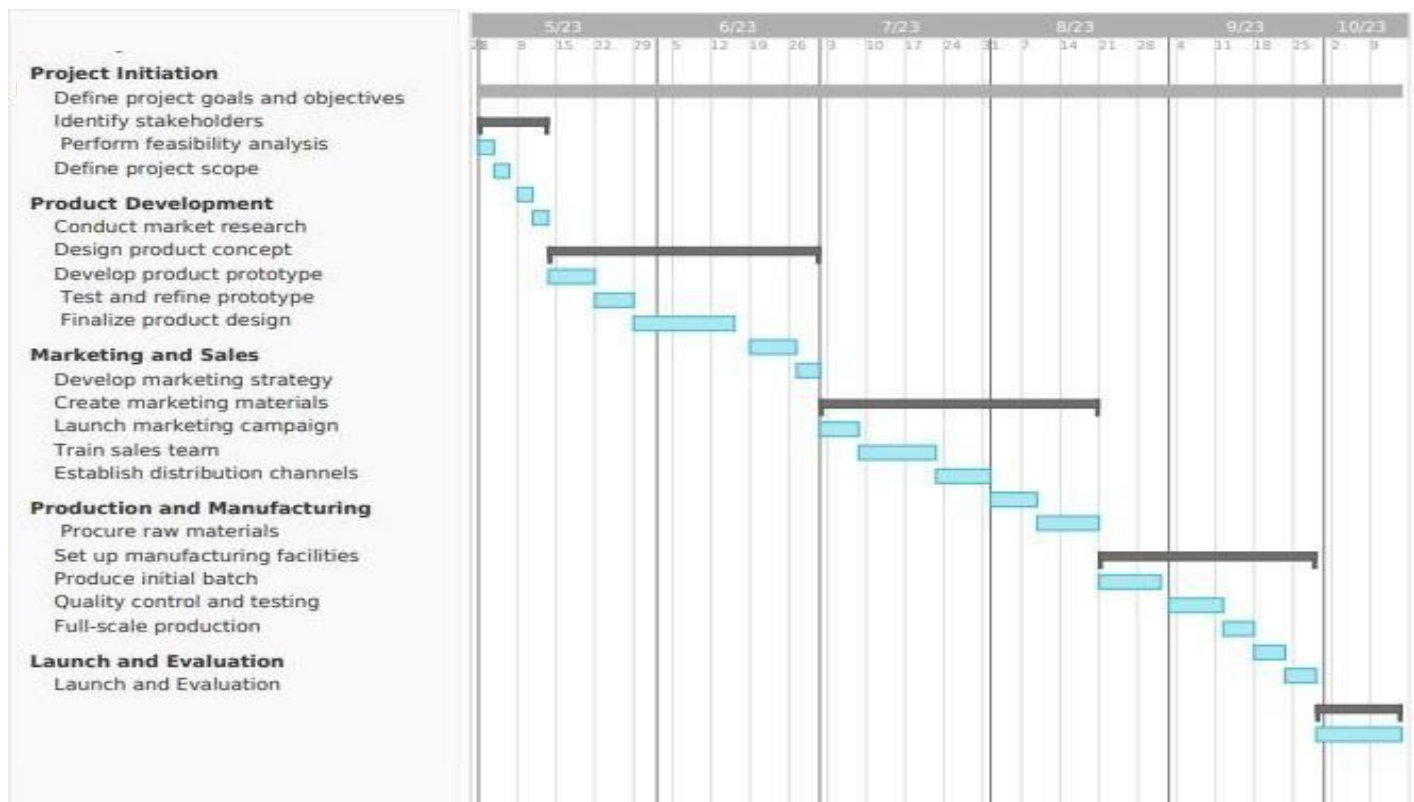
$$FP(\text{estimated}) = \text{count-total} * [0.65 + 0.01 * \sum(F_i)]$$

$$= 57 * [0.65 + 0.01 * 51]$$

= 66.12 function points.

For WBS and Gantt Chart

1. Create different tasks of your project depending on the process model that you have selected for the same.
2. Create a WBS or Task network diagram depicting clearly which tasks can happen in parallel and sequential.
3. Identify team members and allot them to the task
4. Based on the effort estimated in person months, distribute the effort on a 40-20-40 % basis.
5. Prepare a Gantt chart using any PM tool





SHRI VILEPARLE 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)



Conclusion:

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



Academic Year: 2022_23
Software Engineering

Experiment-6

Div: B

Batch: B1

Team Members:

Meet Patel - 60004200104

Kartik Jolapara - 60004200107

Aim: Develop Sequence and Collaboration diagram for the project.

Theory:

Sequence Diagram:

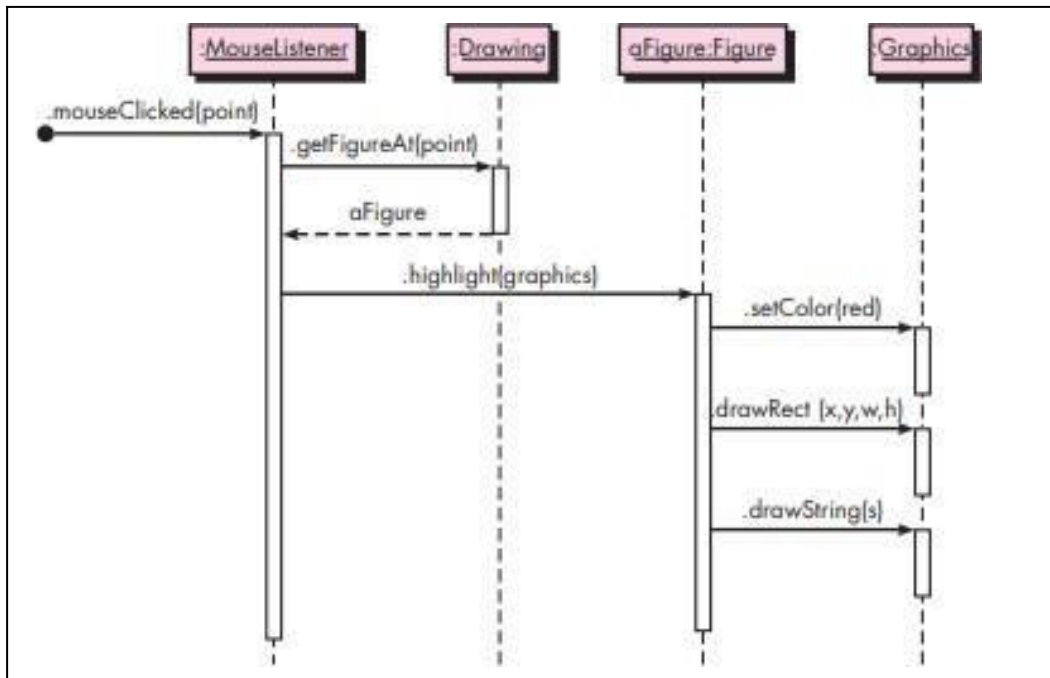
A sequence diagram is used to show the dynamic communications between objects during execution of a task. It shows the temporal order in which messages are sent between the objects to accomplish that task. One might use a sequence diagram to show the interactions in one use case or in one scenario of a software system.

A sequence diagram shows method calls using horizontal arrows from the caller to the callee, labeled with the method name and optionally including its parameters, their types, and the return type.

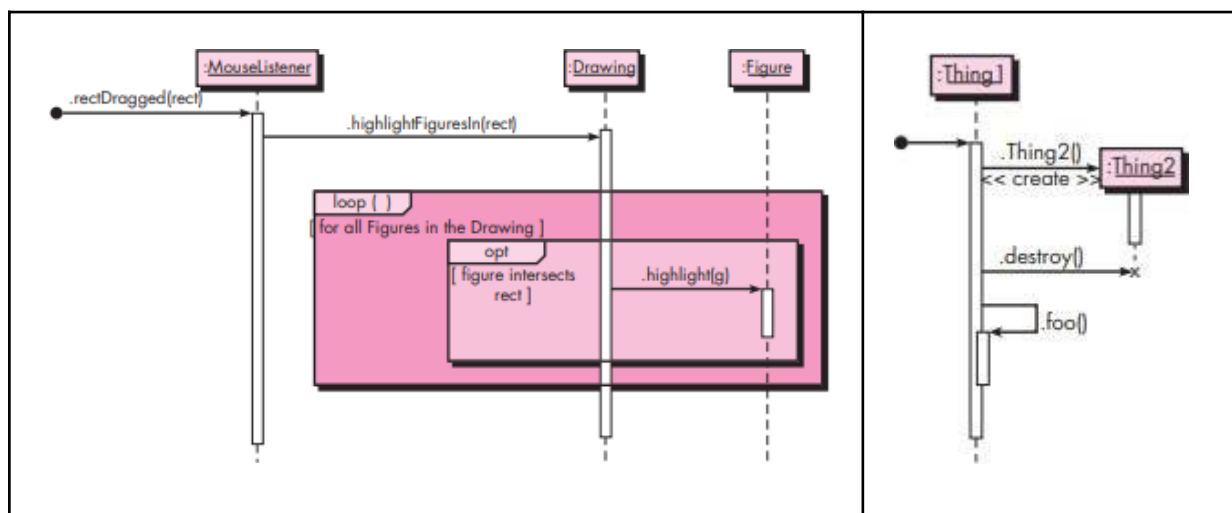
The figure shows a sequence diagram for a drawing program. The diagram shows the steps involved in highlighting a figure in a drawing when it has been clicked. Each box in the row at the top of the diagram usually corresponds to an object, although it is possible to have the boxes model other things, such as classes.

If the box represents an object (as is the case in all our examples), then inside the box you can optionally state the type of the object preceded by the colon. You can also precede the colon and type by a name for the object, as shown in the third box in Figure. Below each box there is a dashed line called the lifeline of the object.

Academic Year: 2022_23



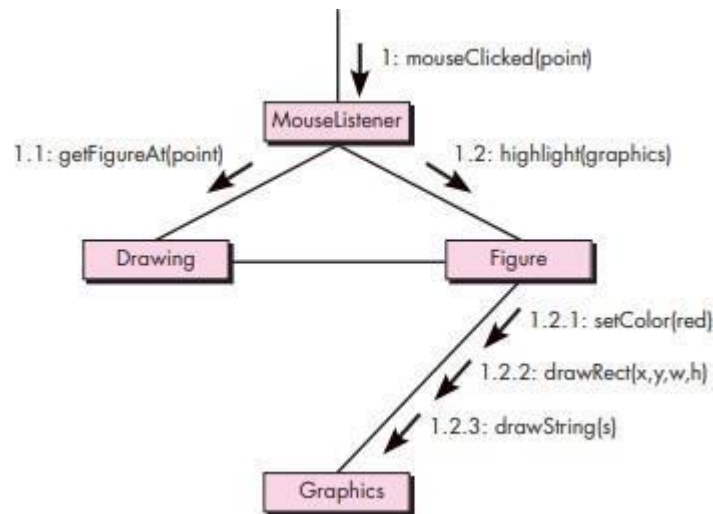
The vertical axis in the sequence diagram corresponds to time, with time increasing as you move downward. The diagram in Figure is very straightforward and contains no conditionals or loops. If logical control structures are required, it is probably best to draw a separate sequence diagram for each case. That is, if the message flow can take two different paths depending on a condition, then draw two separate sequence diagrams, one for each possibility



Academic Year: 2022_23

Collaboration (communication) Diagrams:

A collaboration diagram, illustrated in Figure below, displays the same actions shown in the sequence diagram in Figure.



In a collaboration diagram the interacting objects are represented by rectangles. Associations between objects are represented by lines connecting the rectangles. There is typically an incoming arrow to one object in the diagram that starts the sequence of message passing. That arrow is labeled with a number and a message name. If the incoming message is labeled with the number 1 and if it causes the receiving object to invoke other messages on other objects, then those messages are represented by arrows from the sender to the receiver along an association line and are given numbers 1.1, 1.2, and so forth, in the order they are called. If those messages in turn invoke other messages, another decimal point and number are added to the number labeling these messages, to indicate further nesting of the message passing. In Figure, you see that the **mouseClicked** message invokes the methods **getFigureAt()** and then **highlight()**. The **highlight()** message invokes three other messages: **setColor()**, **drawRect()**, and **drawstring()**. The numbering in each label shows the nesting as well as the sequential nature of each message.



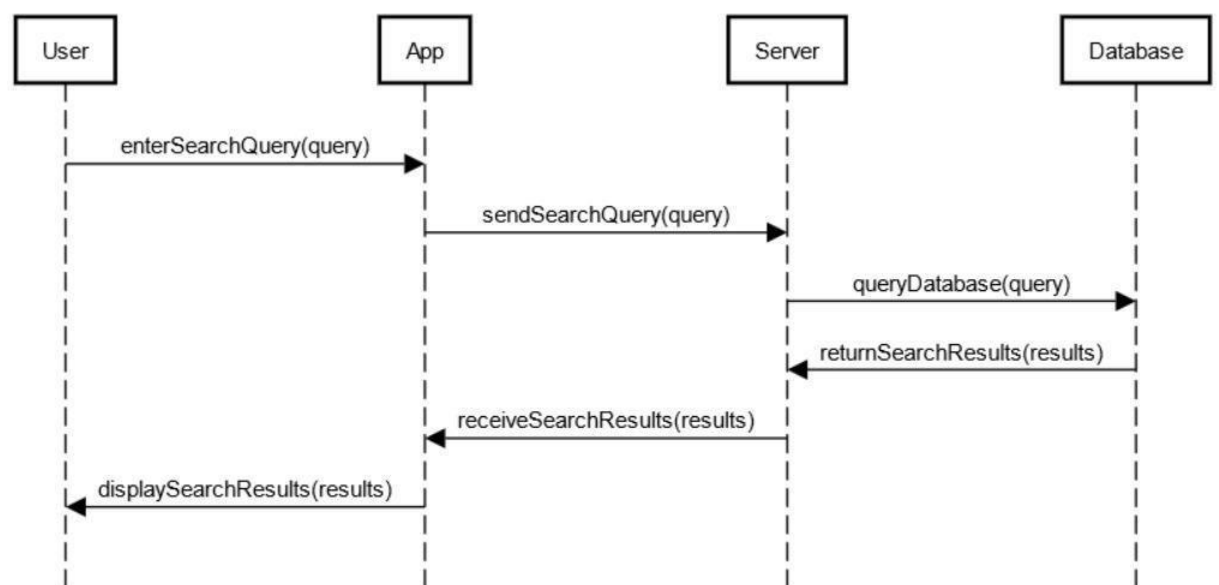
Academic Year: 2022_23

Practical:

For Sequence diagram

1. Identify any functionality involving more than two objects of your case study from Expt 3.
2. Identify the communication between the actors. Represent the same in the form of messages in the sequence diagram.

Let us consider a sequence diagram for the functionality of a user searching for and selecting a news



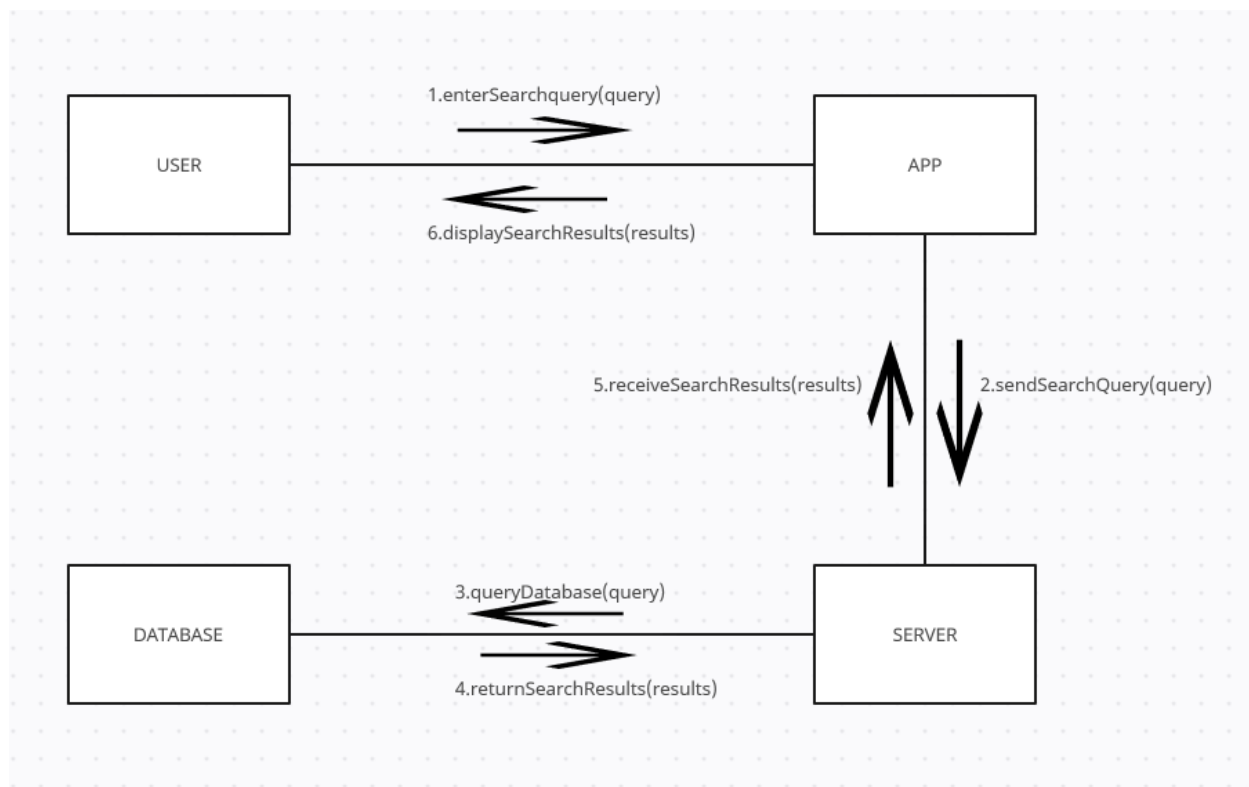


Academic Year: 2022_23

For Collaboration diagram

1. Convert the Sequence diagram for the same functionality into Collaboration diagram using respective notations

Let us consider a collaboration diagram for the functionality of a user searching for and selecting a news



Conclusion:

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



Academic Year: 2022_23
Software Engineering

Experiment-7

Div: B

Batch: B1

Team Members:

Meet Patel - 60004200104

Kartik Jolapara - 60004200107

Aim: Design test scenarios and test cases for your SRS.

Performance:

1. Create test scenario and test cases for your case study
2. Use the following template for the Test Scenario

Project Name	Inshorts			
Reference Document	Software Requirement Specification , Version 1			
Created by	Meet Patel, Kartik Jolapara			
Date of creation	10-Apr-23			
Date of review	17-Apr-23			
Test scenario ID	Requirement-reference document index	Test scenario description	Importance	No. of test cases
TS001	3.2.1	Verify that users can create an account on the platform	High	1
TS002	3.2.2	Verify that users can post news articles/summaries to the platform and that they are securely stored on the database.	High	3



Academic Year: 2022_23

TS003	3.2.3	Verify that users can search for particular news articles	High	2
TS004	3.2.4	Verify that users can watch the news summaries in video format	High	3
TS005	3.2.5	Verify that users can rate and comment on the videos on the platform and that these ratings and reviews are stored on the database.	Medium	2
TS006	3.2.6	Verify that the platform can detect and prevent any attempts at fraud or false news.	High	1

Column #1: Test scenario ID

Every entity in our testing process has to be uniquely identifiable. So, every test scenario has to be assigned an ID. The rules to follow while assigning this ID have to be defined. For the sake of this article we are going to follow the naming convention as: TS(prefix that stands for Test Scenario) followed by ‘_’, module name MI(my Info module of the Orange HRM project) followed by ‘_’ and then the sub section (eg: MIM for My info module, P for photograph and so on)followed by a serial number. An example would be: “TS_MI_MIM_01”.

Column #2: Requirement

It helps that when we create a test scenario we should be able to map it back to the section of the SRS document where we picked it from. If the requirements have IDs we could use that. If not section numbers or even page numbers of the SRS document from where we identified a testable requirement will do.

Column #3: Test scenario description

A one liner specifying ‘what to test’. We would also refer to it as test objective.

Column #4: Importance

This is to give an idea about how important certain functionality is for the AUT. Values like high, medium and low can be assigned to this field. You could also choose a point system, like 1-5, 5 being most important, 1 being less important. Whatever the value this field can take, it has to be pre-decided.

Column #5: No. of Test cases

A rough estimate on how many individual test cases we might end up writing that one test scenario. **For example:** To test the login- we include these situations: Correct username and password. Correct username and wrong password. Correct password and wrong username. So, validating the login functionality will result in 3 test cases.



Academic Year: 2022_23

3. Use the following template for the Test Cases

Test case ID: Unique ID for each test case. Follow some convention to indicate types of test. E.g. 'TC_UI_1' indicating 'user interface test case #1'.

Test Objective: Describe what the test case is actually going to test.

Pre-condition: Any prerequisite that must be fulfilled before execution of this test case. List all pre-conditions in order to successfully execute this test case.

Dependencies: Mention any dependencies on other test cases or test requirement.

Test Steps: List all test execution steps in detail. Write test steps in the order in which these should be executed. Make sure to provide as much details as you can. Tip – to efficiently manage test case with lesser number of fields use this field to describe test conditions, test data and user roles for running test.

Test Data: Use of test data as an input for this test case. You can provide different data sets with exact values to be used as an input.

Expected Result: What should be the system output after test execution? Describe the expected result in detail including message/error that should be displayed on screen.

Post-condition: What should be the state of the system after executing this test case?

Actual result: Actual test result should be filled after test execution. Describe system behavior after test execution.

	Project Name	Inshorts				
	Reference Document	Software Requirement Specification , Version 1				
	Created by	Meet Patel, Kartik Jolapara				
	Date of creation	10-Apr-23				
	Date of review	17-Apr-23				



Academic Year: 2022_23

Test case ID	Test Objective	Precondition	Steps:	Test data	Expected result	Post-condition
TC002_1	Post news articles/videos	User is not logged in	1. Navigate to the upload page 2. Fill the necessary details 3. Click on the "Upload" button	Valid news	User is prompted to login on the platform	Redirect to the login page for login
TC002_2	Post news articles/videos	User is logged in	1. Navigate to the upload page 2. Fill the necessary details 3. Click on the "Upload" button	Valid news	News is uploaded and securely stored on the database	News article and news reels are posted on the platform after verification
TC003_1	Display the searched news	Searched news article is available	1. Navigate to the news page 2. Search for the article	Stable internet connection	Articles is available for reading	User reads the article
TC003_2	Display the searched news	Searched news article is available	1. Navigate to the news page 2. Search for the article	Stable internet connection	Articles is available for reading	Prompt article not available
TC006_1	Comment/rate the video	User is not logged in	1. Navigate to the video/article rating and review page 2. Select the rating and comment 3. Enter the rating and review details 4. Click on the "Submit" button	Valid rating and review details	Rating and review are submitted and recorded	Rating and review are visible on the page and stored in the database
TC006_2	Comment/rate the video	User is logged in	1. Navigate to the video/article rating and review page 2. Select the rating and comment 3. Enter the rating and review details 4. Click on the "Submit" button	Valid rating and review details	User is prompted to login on the platform	Redirect to the login page for login

Conclusion:

Thus, we are able to create test scenarios and test cases for our case study.



A.Y. 2021 – 22

Software Engineering

Experiment-8

Div: B

Batch: B1

Team Members:

Meet Patel - 60004200104

Kartik Jolapara - 60004200107

Aim: Study of Azure Devops

Theory:

Azure DevOps provides developer services for allowing teams to plan work, collaborate on code development, and build and deploy applications. Azure DevOps supports a collaborative culture and set of processes that bring together developers, project managers, and contributors to develop software. It allows organizations to create and improve products at a faster pace than they can with traditional software development approaches.

Azure DevOps provides integrated features that you can access through your web browser or IDE client.

Azure Repos:

Azure Repos is a set of version control tools that you can use to manage your code. Version control systems are software that help you track changes you make in your code over time. As you edit your code, you tell the version control system to take a snapshot of your files. The version control system saves that snapshot permanently so you can recall it later if you need it. Use version control to save your work and coordinate code changes across your team.

Azure Repos provides two types of version control:

1. Git repositories: Git is the most commonly used version control system today and is quickly becoming the standard for version control. Git is a distributed version control system, meaning that your local copy of code is a complete version control repository. These fully functional local repositories make it is easy to work offline or remotely. You commit your work locally, and then sync your copy of the repository with the copy on the server.
2. Team Foundation Version Control (TFVC): Azure Repos also supports Team Foundation Version Control (TFVC). TFVC is a centralized version control system.



A.Y. 2021 – 22

Typically, team members have only one version of each file on their dev machines. Historical data is maintained only on the server. Branches are path-based and created on the server.

Azure Pipelines:

Azure Pipelines automatically builds and tests code projects to make them available to others. It works with just about any language or project type. Azure Pipelines combines continuous integration (CI) and continuous delivery (CD) to test and build your code and ship it to any target.

Continuous Integration (CI) is the practice used by development teams of automating merging and testing code. Implementing CI helps to catch bugs early in the development cycle, which makes them less expensive to fix. Automated tests execute as part of the CI process to ensure quality. Artifacts are produced from CI systems and fed to release processes to drive frequent deployments. The Build service in Azure DevOps Server helps you set up and manage CI for your applications.

Continuous Delivery (CD) is a process by which code is built, tested, and deployed to one or more test and production environments. Deploying and testing in multiple environments increases quality. CI systems produce deployable artifacts, including infrastructure and apps. Automated release processes consume these artifacts to release new versions and fixes to existing systems. Monitoring and alerting systems run continually to drive visibility into the entire CD process.

Azure Boards:

Delivers a suite of Agile tools to support planning and tracking work, code defects, and issues using Kanban and Scrum methods. Azure Boards provides software development teams with the interactive and customizable tools they need to manage their software projects. It provides a rich set of capabilities including native support for Agile, Scrum, and Kanban processes, calendar views, configurable dashboards, and integrated reporting. These tools scale as your business grows.



A.Y. 2021 – 22

Quickly and easily track work, issues, and code defects associated with your project. The Kanban board, shown in the following image, is just one of several tools that allows you to add, update, and filter user stories, bugs, features, and epics.

Azure Test Plans:

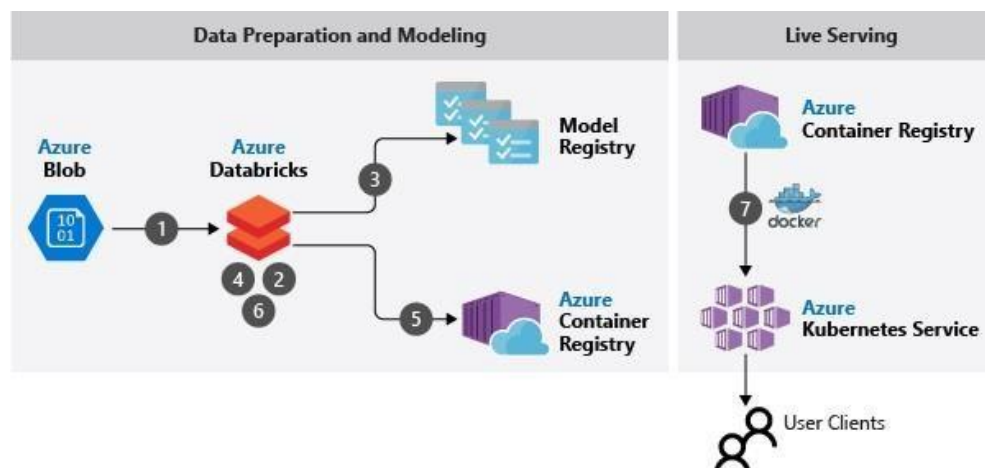
Azure Test Plans provides rich and powerful tools everyone in the team can use to drive quality and collaboration throughout the development process. The easy-to-use, browser-based test management solution provides all the capabilities required for planned manual testing, user acceptance testing, exploratory testing, and gathering feedback from stakeholders.

Azure Artifacts:

Azure Artifacts enable developers to consume and publish different types of packages to Artifacts feeds and public registries such as NuGet.org and npmjs.com. You can use Azure Artifacts in conjunction with Azure Pipelines to deploy packages, publish build artifacts, or integrate files between your pipeline stages to build, test, or deploy your application.

Azure Architecture Solutions:

1. Azure Architecture for Content based Recommendation System:



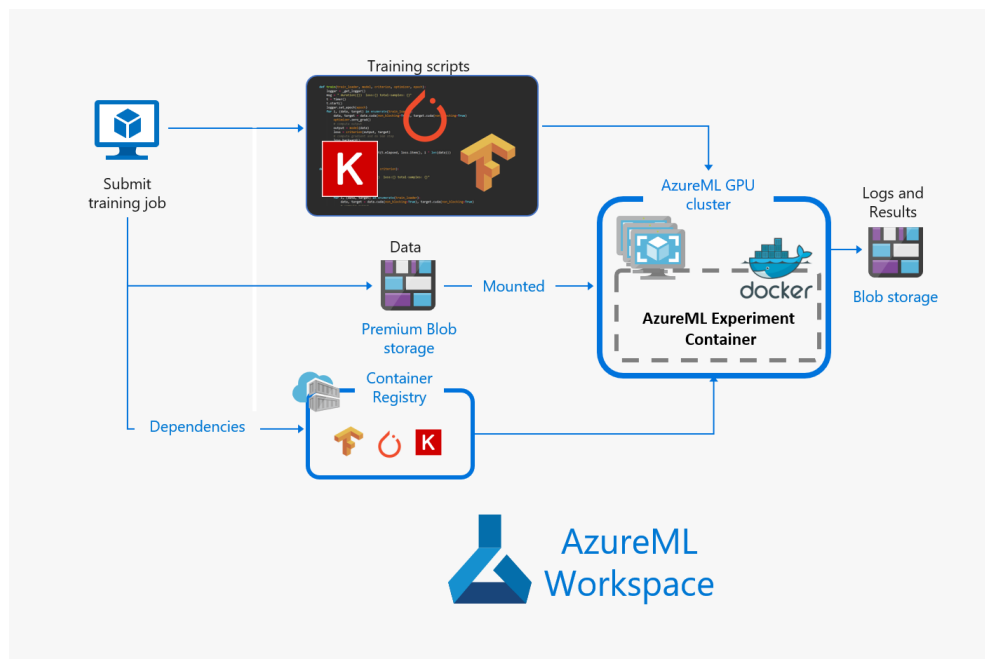
This example scenario covers the training, evaluation, and deployment of a machine learning model for content-based personalization on Apache Spark using Azure Databricks. In this case, a model is trained with a supervised classification algorithm on a dataset containing user and item features. The label for each example is a binary value indicating that the user engaged

A.Y. 2021 – 22

with (for example, clicked) an item. This scenario covers a subset of the steps required for a full end-to-end recommendation system workload. The broader context of this scenario is based on a generic e-commerce website with a front end that serves rapidly changing content to its users. This website uses cookies and user profiles to personalize the content for that user. Along with user profiles, the website may have information about every item it serves to each user.

2. Azure Architecture for Distributed Training Deep learning models:

This reference architecture shows how to conduct distributed training of deep learning models across clusters of GPU-enabled VMs. The scenario is image classification, but the solution can be generalized to other deep learning scenarios such as segmentation or object detection.



Workflow

This architecture consists of the following services:

Azure Machine Learning Compute plays the central role in this architecture by scaling resources up and down according to need. Azure ML Compute is a service that helps provision and manage clusters of VMs, schedule jobs, gather results, scale resources, and handle failures. It supports GPU-enabled VMs for deep learning workloads.

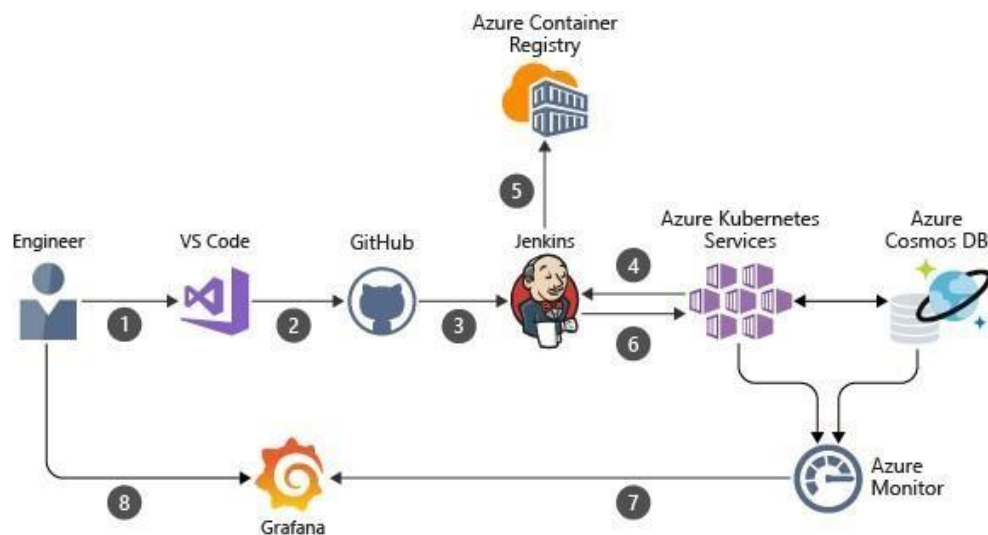


A.Y. 2021 – 22

Standard Blob storage is used to store the logs and results. Premium Blob storage is used to store the training data and is mounted in the nodes of the training cluster using blobfuse. The Premium tier of Blob storage offers better performance than the Standard tier and is recommended for distributed training scenarios. When mounted using blobfuse, during first the epoch, the training data is downloaded to the local disks of the training cluster and cached. For every subsequent epoch, the data is read from the local disks, which is the most performant option.

Container Registry is used to store the Docker image that Azure Machine Learning Compute uses to run the training.

3. Azure Architecture for CI/CD:



Azure Web Apps is a fast and simple way to create web apps using ASP.NET, Java, Node.js, or PHP. Deliver value faster to your customers with a continuous integration and continuous deployment (CI/CD) pipeline that pushes each of your changes automatically to Web Apps.



Software Engineering

Experiment-9

Div: B

Batch: B1

Team Members:

Meet Patel - 60004200104

Kartik Jolapara - 60004200107

AIM :

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

PERFORMANCE :

1. Identify Risks

- a. Refer to the Risk Identification Checklist to be identify the risk
 - i. Product Size Risks
 - ii. Business Impact Risks
 - iii. Customer Related Risks
 - iv. Process Risks
 - v. Technical Issues
 - vi. Technology Risks
 - vii. Development Environment Risks viii. Staff Size and Experience Risks

2. Prepare Risk Table by identifying potential risks and categorizing their impacts as follows Impact Values:

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



Risks	Category	Probability	Impact
Limited content sources	Product Size Risks	45%	1
Low user adoption	Business Impact Risks	52%	2
Lack of personalization	Customer Related Risks	40%	2
Unclear development process	Process Risks	35%	1
Lack of technical expertise	Technology Risks	42%	2
Lack of version control	Development Environment Risks	60%	3

After sorting the above risk table based on impact and probability of risks.

Risks	Category	Probability	Impact
Limited content sources	Product Size Risks	45%	1
Unclear development process	Process Risks	35%	1



Low user adoption	Business Impact Risks	52%	2
Lack of technical expertise	Technology Risks	42%	2
Lack of personalization	Customer Related Risks	40%	2
Lack of version control	Development Environment Risks	60%	3

So the cutoff line for risk table is defined after considering the risks from above tables and their probabilities and by accounting budget, time factor etc.is 40%
cutoff line = 40%.

3. Create a Risk Mitigation, Monitoring and Management plan for the Risks identified in the Risk Table

Risk Information Sheet			
RISK ID :- P01-11	DATE : 30/04/2023	PROB : 45%	IMPACT :- High
Description :- There is a risk of having limited content sources for the news application, which may lead to a lack of diverse and up-to-date news coverage, resulting in reduced user engagement and satisfaction.			



Refinement/Context :-

The risk of limited content sources can arise due to various reasons. It could be because the application has not established partnerships with a wide range of reputable news organizations, publishers, or content providers. It could also be a result of relying on a small number of content sources or not having a robust system to aggregate news from different outlets.

Mitigation/Monitoring :-

1. Regular Content Audits: Conduct periodic audits to evaluate the diversity and quality of content sources. Assess the coverage of different topics, geographical areas, and perspectives. Identify any gaps in coverage and take steps to address them.
2. User Feedback and Analytics: Monitor user feedback, ratings, and reviews related to the content sources. Analyze user engagement metrics to identify patterns that may indicate dissatisfaction with the available content. This can help detect any issues related to limited content sources.

Management:-

1. Content Source Evaluation: Continuously evaluate the performance and relevance of existing content sources. Regularly review agreements with content partners and assess their content quality and reliability. Consider adding or removing sources based on their performance and user feedback.
2. Proactive Content Acquisition: Actively seek out new content sources by monitoring the media landscape, identifying emerging publishers, and conducting research on reliable and diverse news outlets. Explore opportunities to collaborate with new sources to expand the content offering.
3. Continuous Improvement: Regularly update the application's content strategy based on user feedback, market trends, and technological advancements. Keep track of emerging news platforms, social media trends, and content consumption patterns to stay ahead of evolving user needs and preferences.



4. User Engagement and Communication: Maintain open channels of communication with users to gather feedback and address any concerns related to content sources. Encourage users to provide suggestions and recommendations for improving the news content. Communicate any updates or changes in content sources to users transparently.
5. Crisis Response: Develop a contingency plan in case any major content sources become unavailable or unreliable. Identify backup options and alternative sources to mitigate the impact on users and ensure a continuous flow of news content.

Current Status :-

30/04/2023 - Mitigation steps initiated

Originator :- Meet Patel

Assigned :- Kartik Jolapara

Conclusion:

Hence, Risks were identified and a risk table was created and a risk information sheet was made for the same.



A.Y. 2022 – 23

Software Engineering

Experiment-10

Div: B

Batch: B1

Team Members:

Meet Patel - 60004200104

Kartik Jolapara - 60004200107

Aim: Study of Configuration Management using GitHub

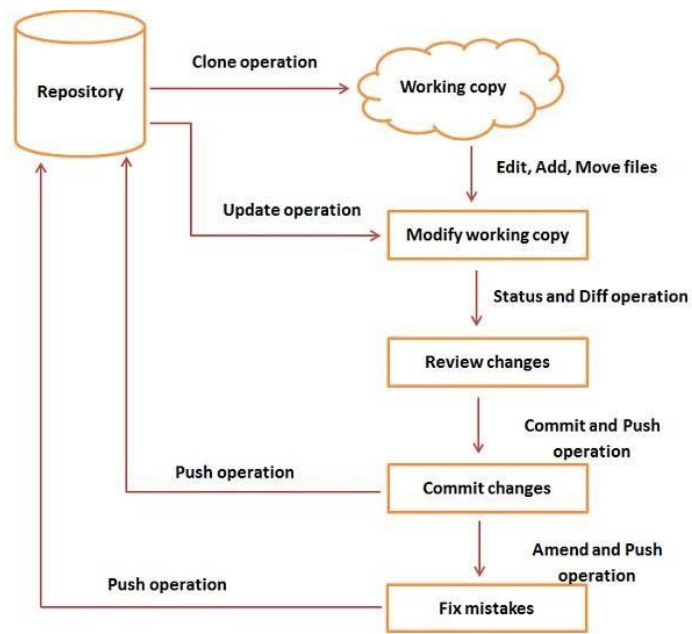
Theory:

Git is a distributed revision control and source code management system with an emphasis on speed. Git was initially designed and developed by Linus Torvalds for Linux kernel development. Git is a free software distributed under the terms of the GNU General Public License version 2.

Git Life Cycle

General workflow is as follows –

1. Clone the Git repository as a working copy.
2. Modify the working copy by adding/editing files.
3. If necessary, update the working copy by taking other developer's changes.
4. Review the changes before commit.
5. Commit changes. If everything is fine, then push the changes to the repository.
6. After committing, if something is wrong, then correct the last commit and push





A.Y. 2022 – 23
Git Life Cycle

1. Creating Git Repository

Initialize a new repository by using **init** command followed by **--bare** option. It initializes the repository without a working directory. By convention, the bare repository must be named as **.git**.

```
[gituser@CentOS ~]$ pwd
/home/gituser

[gituser@CentOS ~]$ mkdir project.git

[gituser@CentOS ~]$ cd project.git/

[gituser@CentOS project.git]$ ls

[gituser@CentOS project.git]$ git --bare init Initialized
empty Git repository in /home/gituserm/project.git/

[gituser@CentOS project.git]$ ls
branches config description HEAD hooks info objects refs
```

```
DJSCE.Student@MUM0922CPU0710 MINGW64 ~
$ pwd
/c/Users/djsce.student

DJSCE.Student@MUM0922CPU0710 MINGW64 ~
$ mkdir project.git

DJSCE.Student@MUM0922CPU0710 MINGW64 ~
$ cd project.git/

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git
$ ls

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git
$ git --bare init
unknown option: --
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
          [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
          [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
          [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
          [--config-env=<name>=<envvar>] <command> [<args>]

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git
$ git --bare init
Initialized empty Git repository in C:/Users/djsce.student/project.git/

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git (BARE:master)
$ ls
HEAD config description hooks/ info/ objects/ refs/

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git (BARE:master)
$ :
```



A.Y. 2022 – 23

2. Generate Public-Private RSA Key Pair

```
User1@CentOS ~]$ pwd
/home/user1

[user1@CentOS ~]$ ssh-keygen
```

```
DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git (BARE:master)
$ pwd
/c/Users/djsce.student/project.git

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git (BARE:master)
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/djsce.student/.ssh/id_rsa):
Created directory '/c/Users/djsce.student/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/djsce.student/.ssh/id_rsa
Your public key has been saved in /c/Users/djsce.student/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:siF8i6a2HRpCT+lpQR3XMg/E7v49crM76cmBZkDz47E DJSCE.Student@MUM0922CPU0710
The key's randomart image is:
+---[RSA 3072]-----+
|      .oo.      |
|      . 0= .    |
|      . ..0=    |
|      ... ..0.  |
|      . +0 =.S+  |
|      . + 0+ *0 = |
|      . =+ + E .. |
|      .o* . .o.o*o |
|      .+.. ..+B*  |
+---[SHA256]-----+
```

3. Adding keys to authorized keys

Suppose there are two developers working on a project. Both users have generated public keys.

Both add their public key to the server by using ssh-copy-id command as given below

```
[user1@CentOS ~]$ pwd
/home/user1

[user2@CentOS ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub
gituser@git.server.com
```




A.Y. 2022 – 23

```
DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git (BARE:master)
$ pwd
/c/Users/djsce.student/project.git

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git (BARE:master)
$ ssh-copy-id -i ~/.ssh/id_rsa.pub
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/c/Users/djsce.student/.ssh/id_rsa.pub"
Usage: /usr/bin/ssh-copy-id [-h|-?|-f|-n|-s] [-i [identity_file]] [-p port] [-F alternative_ssh_config_file] [[-o <ssh -o options>] ..
.] [user@]hostname
-f: force mode -- copy keys without trying to check if they are already installed
-n: dry run -- no keys are actually copied
-s: use sftp -- use sftp instead of executing remote-commands. Can be useful if the remote only allows sftp
-h|-?: print this help
```

4. Push changes to the repository

We have created a bare repository on the server and allowed access for two users. Both users can push their changes to the repository by adding it as a remote.

Git init command creates **.git** directory to store metadata about the repository every time it reads the configuration from the **.git/config** file.

User1 creates a new directory, adds README file, and commits his change as initial commit. After commit, he verifies the commit message by running the **git log** command.

```
[user1@CentOS ~]$ pwd
/home/user1

[user1@CentOS ~]$ mkdir user1_repo

[user1@CentOS ~]$ cd user1_repo/

[user1@CentOS user1_repo]$ git init Initialized
empty Git repository in
/home/user1/user1_repo/.git/

[user1@CentOS user1_repo]$ echo 'TODO: Add contents for
README' > README

[user1@CentOS user1_repo]$ git status -s
?? README

[user1@CentOS user1_repo]$ git add .

[user1@CentOS user1_repo]$ git status -s
A README

[user1@CentOS user1_repo]$ git commit -m 'Initial commit'
```

5. Checking log message by executing the git log command.

```
[user1@CentOS user1_repo]$ git log
```



A.Y. 2022 – 23

```
DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git (BARE:master)
$ pwd
/c/Users/djsce.student/project.git

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git (BARE:master)
$ mkdir user1_repo

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git (BARE:master)
$ cd user1_repo/

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git/user1_repo (BARE:master)
$ git init
Initialized empty Git repository in C:/Users/djsce.student/project.git/user1_repo/.git/

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git/user1_repo (master)
$ echo 'TODO: Add Contents for README'> README

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git/user1_repo (master)
$ git status -s
?? README

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git/user1_repo (master)
$ git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
    add

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git/user1_repo (master)
$ git add .
warning: in the working copy of 'README', LF will be replaced by CRLF the next time Git touches it

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git/user1_repo (master)
$ git status -s
A README

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git/user1_repo (master)
$ git commit -m 'Initial commit'
[master (root-commit) 8f9cd90] Initial commit
Committer: DJSCE Student <DJSCE.Student@SVKMGRP.COM>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+)
create mode 100644 README

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git/user1_repo (master)
$ git log
commit 8f9cd90f5ca19fa2d214245f34ed436830e16c23 (HEAD -> master)
Author: DJSCE Student <DJSCE.Student@SVKMGRP.COM>
Date: Tue May 2 12:24:10 2023 +0530

    Initial commit
```



A.Y. 2022 – 23

6. Commit changes

To commit the changes, he used the git commit command followed by -m option. If we omit -m option. Git will open a text editor where we can write multiline commit message

```
[user2@CentOS project]$ git commit -m 'Implemented  
my_strlen function'
```

```
DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git/user1_repo (master)
$ git commit -m 'Implemented my_strlen function'
On branch master
nothing to commit, working tree clean

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git/user1_repo (master)
$ git log
commit 8f9cd90f5ca19fa2d214245f34ed436830e16c23 (HEAD -> master)
Author: DJSCE Student <DJSCE.Student@SVKMGRP.COM>
Date: Tue May 2 12:24:10 2023 +0530

    Initial commit

DJSCE.Student@MUM0922CPU0710 MINGW64 ~/project.git/user1_repo (master)
$
```

Conclusion:

Git is a version control software that can store different versions on a local machine or can be integrated with remote file management system. We used Git bash cmd to run commands and store the local files on github files management server. We also seen the git log command which stores the the activity log happens on git.

Assignment-1

60004200107

- Q1. Elaborate the task set for creating component level design in oo concepts
- Component level design is an important step in oo software development. It involves creating detailed design for individual components of a software system based on requirement & specifications. Task set involved in creating a component level design:
 - 1) Identify all design classes that correspond to the problem domain. Using requirement and architecture model, each analysis class and architectural component is elaborated.
 - 2) Identify all design classes that correspond to the infrastructure domain: These classes are not described in requirement model and are inherited from the architecture model. As we noted it includes GUI components, operating system components and object and data management components.
 - 3) Elaborate all design classes that are not defined as reusable components. Elaboration requires that all interfaces, attributes, and operations necessary to implement the class as described in detail design hierarchy must be considered as this task is conducted.

(b) Specify message details when classes or components collaborate.

The requirements model makes use of a collaboration diagram to show how analyst's classes collaborate with one another. As component level ~~the~~ design proceeds, it is sometimes useful to show the details of these collaborations. Although this is optional it can be used as a precursor to specification.

(b) Identify appropriate interfaces for each component, in essence operations defined for the design class and categorized into one or more abstract class every operation within the abstract classes.

(c) Elaborate attributes and derive data types and data structures required to implement them. If an attribute appears repeatedly across a number of design classes and it is a relatively complex structure.

(d) Describe processing flow with in each operation in details. This may be accomplished using programming languages pseudocode, or with a control activity diagram.

A) Describe persistent data sources (databases and files) and identify classes to manage them. Database & files normally transcend the design description of an individual component. In most cases,

these points for data structures.

- 5) Develop and elaborate behavioural representations for a class or component:

UML state diagrams were used as part of the requirements made to represent the externally observable behaviour of the system. During component level design it is sometimes necessary.

- a) Elaborate deployment diagrams to provide additional implementation detail.

Deployment diagrams are used as part of architecture design and are represented in descriptor format. Deployment diagrams can be elaborated to represent the location of key packages of components. However, components are not represented individually.

- 4) Refactor every component level design representation and always alternatives.

It is an alternative process, first component model will not be complete, consistent or accurate. They do not suffer from tunnel vision. Develop alternatives and consider each carefully.

Q2 Explain golden rules of user interface design the following three rules:

- (i) Place user in control
- (ii) Reduce user's memory load.
- (iii) Make interface consistent.

(i) Place user in control

- Define interaction models such a way that does not force a user into unnecessary undesired actions
- Provide flexible interaction
- Allow user interaction to be interruptible and undoable.
- Streamline interaction as skill levels advance and allow the interaction to be customized.

(ii) Reduce user's memory load.

- Reduce demand on short term memory
- Establish meaningful defaults.
- Define shortcuts that are intuitive.
- The visual layout of interface should be based on real world metaphors

(iii) make interface consistent

- Allow user to put the current task into meaningful context.
- Maintain consistency across a family of applications
- If past interactive models have created user expectations, do not make things worse than is a compelling reason to do so.

SE - Assignment 2

Q1. Explain equivalence partitioning and boundary value analysis.

Equivalence partitioning & boundary value analysis are two techniques used in software testing to identify test cases and reduce the number of test cases to test a software system.

(i) Equivalence partitioning

It is used to divide a range of input values into smaller and more manageable subsets that are more likely to behave similarly. Input values are divided into two groups or partitions based on their equivalence or similarity. The idea is that if one test case from a partition is successful, all others would be as well.

(ii) Boundary value Analysis

It is a testing technique used to identify errors or defects that occur at or around the boundaries of input values. Input values are chosen from the edges of their ranges under the behaviour of the system may change. The idea is that if a system is functioning correctly at boundaries it will likely function correctly within that range.

Q2. with example, explain OAT

- Operational Acceptance Testing (OAT) is a type of testing that focuses on verifying whether a software application can cooperate in its intended production environment.
- OAT is conducted after completion of a system and user acceptance testing and before software is deployed into production.
- Example: A company has developed a new e-commerce website and they want to ensure that it can handle high traffic loads and transactions without issues.
- They may conduct OAT to verify that the website's servers, network infrastructure and software components can handle the expected load and perform well under stress.
- During OAT, the company may simulate realistic production scenarios such as high traffic loads, simultaneous transactions and data backups.

Q3. Explain version control in SCM

- Version control is a critical aspect of software configuration management (SCM) that helps developers track changes made to source code, documentation and other files over time.
- It is a process that allows multiple developers to work on the same codebase, keep track of changes and collaborate effectively.
- At its core, version control involves creating a repository which is a centralised or distributed system for storing files and recording changes.
- The VCS keeps track of all changes made to them and when they were made.
- There are two main types of VCS
 - (i) Centralised - They have a single central repository where all changes are stored. Developers must keep track of changes and then check them back in. Examples of CVS include Subversion and Perforce.
 - (ii) Distributed VCS - create a local copy of the entire repository on each developer's machine. This means the developers can work independently without need for checking out each time.