**MODULE 4**

Case Study 1: Design Workflow   
you're a lead creator at “ITSYBITSY Creator ORG,” a cutting-edge plan organization known for its inventive arrangements and consistent execution. The company has as of late taken up a challenging extend to update the site of an unmistakable worldwide design brand, "MACOART." The client has tall desires for the modern website's aesthetics, usefulness, and client encounter.  
Your group comprises of three skilled creators with diverse plan specialties and approaches. Mac is a master in moderate and cutting-edge plan, Alex is gifted in making intelligently and immersive encounters, and John exceeds expectations at consolidating creative components into her plans.  
As the extend advances, your group faces a noteworthy predicament concerning the plan workflow. Each architect contains a special vision for the site, and they emphatically accept their approach will best adjust with MACOART ‘s brand personality and fascinate the target gathering of people.  
Question:  
How will you handle the plan workflow predicament at ITSYBITSY Maker ORG to form a cohesive and extraordinary site for MACOART? Create a key arrange that leverages each designer's qualities whereas guaranteeing a bound together plan course for the extend.  
Consider the taking after angles in your arrangement:  
• How will you encourage collaboration and communication among the originators to cultivate imaginative synergy?  
• What strategies will you use to get it MACOART's brand character and target gathering of people way better?  
• Outline a organized handle that permits each architect to contribute their special point of view whereas guaranteeing a cohesive client involvement.  
• What steps will you take to resolve clashes and contradictions that will emerge amid the plan handle?  
• How will you oversee the venture timeline and apportion assets successfully to meet the client's desires and due date?  
  
Answer:  
Making a cohesive and extraordinary site for MACOART requires a well-thought-out plan workflow that saddles the qualities of each originator whereas advancing collaboration and cooperative energy among the group.  
Collaboration and Communication:  
To cultivate inventive collaboration, standard group gatherings and conceptualizing sessions will be organized. Amid these sessions, each creator can show their thoughts, and the group can collectively choose on the most excellent plan heading. Utilizing computerized collaboration devices will empower consistent communication, sharing of plan concepts, and real-time criticism.  
  
  
Understanding MACOART's Brand and Gathering of people:  
In-depth investigate will be conducted to get it MACOART's brand character, target group of onlookers, and their inclinations. Studies, client interviews, and competitor investigation will give profitable experiences to adjust the plans with the client's vision.  
Organized Plan Handle:  
An organized prepare will be executed to guarantee a cohesive client encounter. The extend will kick off with disposition sheets and fashion guides made collaboratively, joining components from each designer's skill. Later, wireframes and models will be created collaboratively to coordinated intelligently components, creative touches, and advanced aesthetics consistently.  
Strife Determination:  
Within the case of conflicts, a majority rule approach will be received, permitting each architect to voice their suppositions and reach an agreement. When essential, the lead originator will make the ultimate choice, considering the project's destinations and client necessities.  
Extend Timeline and Asset Administration:  
To meet the client's desires and due date, a point by point extend timeline will be built up, sketching out particular points of reference and duties. Each designer's mastery will be designated to pertinent assignments, guaranteeing effectively utilize of assets.  
  
By grasping collaboration, understanding the client's needs, actualizing an organized plan handle, settling clashes, and overseeing assets successfully, Creative Sprint will convey an outwardly captivating and user-friendly site that hoists MACROART's online nearness to modern statures.

Case Study 2: Refactoring Challenge  
You're a computer program designer at "IT Organization," an eminent innovation company known for its imaginative computer program items. One of the company's lead applications, "Data Analyst," has been within the showcase for a few a long time. In any case, with quick progressions in innovation and changing client desires, Data Analyst’s codebase has gotten to be obsolete and challenging to preserve. The application's core functionality is solid, but the codebase lacks modularity, contains legacy components, and has several instances of code duplication. As a result, adding new features and fixing bugs has become increasingly time-consuming, and the risk of introducing new issues is high.  
Your chief has entrusted you with driving a refactoring activity for Data Analyst. The objective is to move forward the code quality, upgrade practicality, and clear the way for future include improvement. You've got a team of talented engineers with mastery completely different programming dialects and ideal models.  
Question:  
Create a comprehensive refactoring procedure for the Data Analyst application. Layout the steps you'd take to refactor the codebase whereas guaranteeing negligible disturbance to the existing usefulness and following to best hones.  
Consider the following aspects in your solution:  
• How will you conduct a code survey and distinguish the zones that require refactoring?  
• What approach will you take to prioritize the refactoring assignments and oversee the workload effectively?  
• Describe the strategies you may utilize to guarantee code seclusion, diminish duplication, and make strides by and large viability.  
• How will you handle potential dangers and challenges related with refactoring, such as presenting unused bugs or influencing client involvement?  
• What measures will you put in put to screen the effect of refactoring on application execution and client fulfilment?  
Answer:  
Refactoring the Data Analyst application requires an orderly approach that equalizations code change with keeping up existing usefulness. Here's a comprehensive refactoring technique:  
Code Survey and Distinguishing proof of Refactoring Ranges:  
Perform an exhaustive code survey to distinguish parts of the codebase that display destitute plan, over the top complexity, or tall coupling. Utilize robotized code examination instruments to distinguish code smells and potential ranges for change.  
Prioritization and Workload Administration:  
Prioritize refactoring errands based on criticality and affect. Address regions that have a tall probability of causing issues or obstructing future improvement to begin with. Make a guide with well-defined turning points to oversee the workload effectively.  
  
Measured quality and Duplication Decrease:  
Present a secluded plan by breaking down solid components into littler, more reasonable modules. Empower the utilize of plan designs and best hones to advance code reusability and practicality. Address code duplication by extricating common usefulness into reusable capacities or classes.  
Chance Moderation:  
Sometime recently making changes, make comprehensive test suites to cover existing usefulness. Receive a test-driven advancement (TDD) approach, where unused tests are composed some time recently refactoring. Conduct incremental refactoring to play down the chances of presenting unused bugs. Actualize nonstop integration and visit code surveys to capture issues early.  
Execution Checking and Client Fulfilment:  
Set up execution benchmarks to survey the effect of refactoring on application speed and asset utilization. Screen client criticism and fulfilment amid and after the refactoring prepare to recognize potential issues and client encounter enhancements.

Case Study 3: UML Object-Oriented Analysis Challenge  
You've got been contracted as a software architect at "Tenhoff," a startup that's creating a progressive smart Phone mechanization framework called "Smart Life." The framework points to supply property holders with progressed control over different domestic gadgets, such as lighting, security cameras, indoor regulators, and amusement frameworks, through a centralized portable application.  
As the venture advances, you realize they require for a comprehensive UML (Bound together Modelling Dialect) object-oriented examination to guarantee a clear understanding of the system's prerequisites, functionalities, and intuitive among different co

Question:  
Perform a UML object-oriented investigation for the "Smart Life" shrewd domestic robotization framework. Create nitty gritty UML graphs that outline the system's engineering, its centre functionalities, and the connections between diverse classes and components.  
Consider the taking after perspectives in your arrangement:  
• Illustrate the different utilize cases for Smart Life, counting client intuitive, gadget controls, and framework settings.  
• Design a comprehensive course graph that speaks to the centre substances and their qualities, strategies, and connections inside the Smart Life framework.  
• Develop a grouping graph that exhibits the stream of intelligent between clients and the Smart Life framework for a particular situation, such as turning on the lights remotely.  
• Create a state machine chart for a chosen domestic gadget (e.g., a shrewd indoor regulator) that illustrates its diverse states and the moves between them.  
• Design a movement chart to layout the method of setting up a modern savvy gadget inside the Smart Life framework.  
Answer:  
Utilize Case Graph:  
The utilize case graph grandstands the essential performing artists (e.g., Mortgage holder, Keen Gadgets) and their intuitive with the Smart Life framework. Utilize cases incorporate functionalities like controlling gadgets, designing settings, and accepting cautions.  
Lesson Chart:  
The lesson graph diagrams the centre substances of the framework, counting Property holder, Smart Device, Smart Light, Smart Thermostat, Security Camera, and others. It outlines their qualities (e.g., gadget ID, gadget title, status), strategies (e.g., turn on (), turnoff (), adjust Temperature ()), and connections (e.g., affiliation, conglomeration, legacy).  
Arrangement Graph:  
The grouping chart illustrates the intelligent between the Property holder and the Smart Life framework when turning on the lights remotely. It incorporates messages such as "Mortgage holder demands light on ()" and "Smart Light sends status 'ON' to Property holder."  
  
State Machine Chart:  
The state machine chart for the Smart Thermostat delineates distinctive states like "Warming," "Cooling," and "Sit out of gear." Moves are activated based on temperature changes, client settings, or framework reactions.  
Action Graph:  
The movement graph grandstands the step-by-step handle of setting up an unused Savvy Gadget inside the Smart Life framework. It incorporates exercises such as "Interface Gadget to Wi-Fi," "Enrol Gadget with Smart Life App," and "Total Gadget Arrangement."

**MODULE 5:**

Case Study 1:   
  
Company X is a rapidly expanding e-commerce platform with many products and customers. They must efficiently store product information, customer data, order history, and transaction details.  
  
Challenges:  
  
Scalability and large transaction volume requirements.  
  
The ability to adapt to changing data structure.  
  
Complex joins and aggregations can be queried efficiently.  
  
Maintaining the consistency and integrity of data.  
  
Solution:  
  
Non-Relational Database Option (MongoDB): Provides scalability, data schema flexibility, and faster development cycles.  
  
Criteria for Decision:  
  
Scalability: NoSQL databases are more easily horizontally scalable than relational databases.  
  
Relational databases are better for complicated queries, but NoSQL databases are better for basic queries on massive datasets.  
  
Because of their flexible schema, NoSQL databases allow for faster iterations.

Case 2:   
Enterprise Z is developing a content management system that will handle a variety of material kinds such as articles, photos, and videos.  
• Creating an effective data model to manage a variety of content kinds.  
• Improving query performance for faster content retrieval.  
• Ensuring data consistency across several content kinds.  
• Adapting to changes in content structure.  
Solution:  
1. MongoDB Non-Relational Database Option: Use a document-based model with embedded or linked content.  
Considerations:  
1. Indexes: Select appropriate indexes to improve query performance.  
2. Syntax: SQL for MySQL, and MongoDB's versatile JSON-like syntax.  
ACID transactions in relational databases; eventual consistency in NoSQL databases.

**MODULE 6:**

Case Study 1:  
<http://demowebshop.tricentis.com/>  
Module: REGISTRATION, LOGIN & BUY ITEMS  
Project Overview:  
Demo web shop is the sample online application for buying items It supports various functionalities such as user registration, login, searching for a product, add items to cart and payments. It is developed by Tricentis. It supports multiple browsers such as IE, Firefox, Chrome.  
In scope Functionality:  
Demo webshop provides following functionalities to the end user :  
1. Registration Page :  
Validate the Firstname, Lastname, Email, Password, Retype password on the page.  
2. Login  
a. Email : validate all fieldsa with valid and invlaid data.  
b. Password : validate with all options - valid and invlaid; the forgot password link, Sign-up link  
3. Search an item   
a. Search Product on various categories - Books, Electronics items, Apparels and shoes, Digital downloads, and Jewellary  
b. View the products availability  
c. Add to cart  
4. Buy items  
a. Enter shipping address  
b. Enter billing address  
c. Enter the payment details  
d. Enter the mode of delivery  
e. Discount coupons   
5. Confirm Booking   
a. Download and acknowledge the receipt  
  
Out of Scope Functionality:  
1. Cancel the order  
2. Any other functionalities  
  
  
Question:  
Implement above requirements with TestNG   
Create Assertion  
Create Parameterization  
Create TestNG report

Case Study 2:  
<https://opensource-demo.orangehrmlive.com/>  
  
Project Overview:  
OrangeHRM is the most popular open source human resource management (HRM) Web Based software application. Orange HRM, being browser based, require no clients to be installed. It has easy- to- use interface, is customizable to any length, and is affordable. The platform provides stability for performing rigorous Testing.  
  
In scope Functionality:  
Login with below credentials :  
1. User should be able to Login to Orange HRM by entering valid credentials on Login Panel.  
  
User Name:Admin  
Password:Admin123  
  
B. The PIM Module  
1. The PIM Module should maintains all relevant employee related information, including different types of personal information, detailed qualifications, work experience, job related information etc.  
2. The Admin should be able to Configure optional/custom fields, using Configuration Tab under PIM Module.  
3. The Admin should be able to View all employee details  
4. The Admin should be able to Add employee on the list.  
5. The admin should be able to Generate employee report.   
6. The ESS-Supervisor should be able to View his personal details as well as his/her subordinates.  
7. The ESS-Employee can view his personal details under the ‘My Info’ Module.   
  
  
Question:  
Implement above requirements with JUNIT   
Create Assertion  
Create Parameterization  
Create TestNG report