**Water Fall Model Documents**

**Waterfall Project2 – Part -2/2 – 100 Marks - Pass 60 %**

**Part 2/2 Evaluation**

**Document 6 - Please prepare a use case diagram, activity diagram and a use case specification document.**

Answer – **Use Case Diagram –**

**Use case Specification Document**

1. **Use Case Name : Check Investor Portfolio**

**Brief Descriiption**-

The User is able to look at the investors current Portfolio

**Actors –**

Sales employee / Operation Employee.

**Flow of Event**

**Basic Flow –**

The use case begins when the user clicks on the ‘investor Enquiry’ Tab.

System displays the Search options (Pan, Mobile number, Email ID, Account number, investors name, etc.)

User enters any one of the above data and clicks on search option.

System Displays a pop up tab of Folio Number for existing details

User clicks on Folio number.

System display the Investor folio page.

User clicks on the valuation tab to look for Current valuations.

**Alternate Flow-**

User cancels the query-

At any point of the if the user signals the system that he/she wishes to cancel the Search.

The use case ends.

User enters an incorrect data-

The use case begins when the user clicks on the ‘investor Enquiry’ Tab.

System displays the Search options (Pan, Mobile number, Email ID, Account number, investors name, etc.)

User enters the data that is not available with the system.

System displays the message ‘Please enter valid data’.

User clicks on ok button

Use case ends.

**Special Requirement**

NA

**Precondition**

User is logged in to the system

There is active network to the system.

**Post condition –**

 Successful Post-Condition

User is able to look at investors current valuation and invested amount

Unsuccessful Post condition

System will display page to enter investors data.

1. **Use Case Name: Check Distributors data.**

**Brief Descriiption**-

The User is able to look at the Distributors data

**Actors –**

Sales employee / Operation Employee.

**Flow of Event**

**Basic Flow –**

The use case begins when the user clicks on the ‘Distributors Details’ Tab.

System displays the Search options (Pan, Mobile number, Email ID, Arn Code, Distributors name, etc.)

User enters any one of the above data and clicks on search option.

System Displays a pop up tab of ‘ARN code’ and ‘Name of distributors’ for existing details

User clicks on ARN Code.

System display the Distributor details page.

**Alternate Flow-**

User cancels the query-

At any point of the if the user signals the system that he/she wishes to cancel the Search.

The use case ends.

User enters an incorrect data-

The use case begins when the user clicks on the ‘Distributor Details’ Tab.

System displays the Search options (Pan, Mobile number, Email ID, Distributors name, Arn Code etc.)

User enters the data that is not available with the system.

System displays the message ‘Please enter valid data’.

User clicks on ok button

Use case ends.

**Special Requirement**

NA

**Precondition**

User is logged in to the system

There is active network to the system.

**Post condition –**

 Successful Post-Condition

User is able to look at distributors details.

Unsuccessful Post condition

System will display page to enter Distributors data.

1. **Use Case Name : Update the details /Create Query**

**Brief Descriiption**-

The User is able Update the details

**Actors –**

Operation Employee.

**Flow of Event**

**Basic Flow –**

The use case begins when the user Clicks on the service tab in Investors details or Distributors details.

System displays the pop up page which displays tabs for type of query, call type, Supporting Documents and issue with (Registrar/Amc)

User enters the data that is to be updated .

System Displays the Scanner button to upload the supporting document.

User Upload the supporting Document if Necessary or Press skip button.

Use case ends

**Alternate Flow-**

User cancels the query-

At any point of the if the user signals the system that he/she wishes to cancel .

The use case ends.

**Special Requirement**

NA

**Precondition**

User is logged in to the system

User has entered in Investors page or Distributors Page

There is active network to the system.

**Post condition –**

 Successful Post-Condition

System Displays the Success message and transaction number

Unsuccessful Post condition

System will display error massage and distributor page or investors page.

1. **Use Case Name : Track Query**

**Brief Descriiption**-

The User is able to track Query

**Actors –**

Sales employee / Operation Employee.

**Flow of Event**

**Basic Flow –**

The use case begins when the user Clicks on the contact tab in Investors details or Distributors details.

System expands the contact button and displays all the query that are issued.

User looks for the process of the query.

User clicks on the transaction ref id so they can look on the process.

User can also click on image icon to look at the documents uploaded.

**Alternate Flow-**

User cancels the query-

At any point of the if the user signals the system that he/she wishes to cancel the Search.

The use case ends.

**Special Requirement**

NA

**Precondition**

User is logged in to the system

There is active network to the system.

User is in the Investors or distributors tab

**Post condition –**

 Successful Post-Condition

User is able to look at Query under process and its details

Unsuccessful Post condition

System will display error massage and distributor page or investors page.

1. **Use Case Name : Download Investors statement.**

**Brief Descriiption**-

The User is able to download the investors statement.

**Actors –**

Sales employee / Operation Employee.

**Flow of Event**

**Basic Flow –**

The use case begins when the user Clicks on the “statement” Investors details.

System Displays new tab for statement.

User clicks on the period or any default period.

User clicks on the mail or default for the statement they want in.

User click on view button

System displays authenticate button or a go through.

If user click authenticate button otp is sent to investors mobile number after entering otp system download the statement.

**Alternate Flow-**

User cancels the query-

At any point of the if the user signals the system that he/she wishes to cancel the Search.

The use case ends.

User enters an incorrect date-

User enters the incorrect date .

System display the error message

User enters the valid date.

**Special Requirement**

NA

**Precondition**

User is logged in to the system

There is active network to the system.

User is in investors tab.

**Post condition –**

 Successful Post-Condition

User is able to Download the statement.

Unsuccessful Post condition

System will display error massage and display the statement tab.

**Activity Diagram –**

**Document 7- Screens and pages**

Please follow the following steps to create the mock-ups

1. Kindly use balsamic or Axure.

2. Always start with a home page of an application.

3. Take a feature and follow it to the end

a. Eg: Home page of SCRUM Foods

b. Select Login- Create a login page

c. Let’s assume, you want to search a restaurant

d. Search page- Type the restaurant name and select the dish

e. Add to cart page

f. Payment page

g. Logout page













**Document 8- Tools-Visio and Axure Write a paragraph on your experience using Visio and Axure for the project.**

Visio: I have used Microsoft Visio to draw use case diagram and activity diagram. In Visio, there are various statistic and dynamic diagrams. For drawing use case diagram, select UML use case where we can find various diagrams such as use case, actor, uses, communication lines, extends etc. For drawing activity diagram, select UML activity where we can find various diagrams such initial state, final state, action state, control flow, decision box etc.

Axure: I have used Axure for prototyping the website. In Axure, we can create multiple webpages as per requirement. There are various icons for various purposes. We can select suitable icons/images to represent various activities. Once prototype is ready same can be used to represent to the stakeholder.

**Document 9- BA experienceMy experience as BA in following phases:**

1.Requirement gathering:

Identifying Stakeholders- I identified who the stakeholders are. These are individuals or groups who have an interest in the project, such as clients, end-users, managers, and developers.

 Conducting Interviews: Interview with stakeholders helped to gather information about their requirements, expectations, and concerns. This is an interactive process that allows for direct communication and clarification of any ambiguities.

Surveys and Questionnaires: Distribute surveys or questionnaires to a larger audience to gather more widespread opinions and preferences. This method is useful for collecting quantitative data.

Observation: Observe the current processes or systems in action to identify potential improvements or issues. This can provide valuable insights into how the new system should function.

Prototyping: Create prototypes or mockups of the system to give stakeholders a visual representation. This can help in clarifying requirements and identifying any misunderstandings early in the process.

Document Analysis: Review existing documentation, such as business processes, policies, and procedures, to extract relevant information that can contribute to the understanding of requirements.

Brainstorming: Facilitate brainstorming sessions with stakeholders to generate ideas and explore potential requirements. This can encourage creative thinking and collaboration.

Workshops and Focus Groups: Conduct workshops or focus groups to bring together stakeholders for in-depth discussions. This allows for collaboration and consensus building among participants.

Use Cases and User Stories: Develop use cases or user stories to describe specific interactions and scenarios that users will have with the system. This helps in detailing functional requirements.

Prioritization - Prioritize requirements based on their importance and impact. This helps in focusing on essential features and functionalities, especially in cases where resources are limited.

Documentation: Document all gathered requirements in a clear and organized manner. This documentation serves as a reference throughout the project and is essential for communication among team members.

2.Requirement Analysis:

Validation of Requirements: Verify the accuracy and correctness of the gathered requirements. Ensure that they align with the overall goals of the project and meet the expectations of stakeholders.

Consistency Check: Examine the requirements for consistency within the document and across different sources. Ensure that there are no conflicting or contradictory statements.

Completeness Check: Verify that all necessary requirements are documented. Ensure that there are no gaps or missing elements that could lead to misunderstandings during the development process.

Feasibility Analysis: Evaluate the feasibility of implementing the identified requirements. Assess technical, operational, and economic feasibility to determine if the proposed solution is viable.

Prioritization: Review the prioritization of requirements established during the gathering phase. Confirm that high-priority features are still relevant and that low-priority items can be deferred or eliminated.

Risk Analysis: Identify and analyze potential risks associated with the requirements. Assess the impact of these risks on the project and develop strategies to mitigate or manage them.

Clarification and Detailing: Clarify ambiguous or unclear requirements. Add necessary details to make the requirements more explicit and understandable for the development team.

Traceability: Establish traceability between requirements and other project artifacts, such as design documents, test cases, and code. This helps ensure that each requirement is addressed and tested appropriately

Prototyping: Create prototypes or models to validate and refine the understanding of requirements. Prototypes can be particularly useful for complex or innovative features.

Feedback Loop with Stakeholders: Engage in a feedback loop with stakeholders to ensure that the refined requirements still meet their expectations. This iterative process helps in achieving a shared understanding of the project's objectives.

Change Control: Implement a change control process to manage and track any changes to the requirements. Changes should be carefully evaluated for their impact on the project's scope, timeline, and resources.

Documentation Update: Update the requirement documentation to reflect any changes, additions, or clarifications made during the analysis phase. Ensure that the documentation remains are liable reference for the development team

3.Design:

Architectural Design: Define the overall system architecture, including major components, modules, and their interactions. This step establishes the framework for the ensure system.

High-Level Design: Develop a high-level design that outlines the structure of the system without delving into the specifics of individual components. This design typically includes data flow diagrams, system flow charts, and other modeling techniques.

Detailed Design: Break down the high-level design into detailed specifications for each component or module. This involves defining data structures, algorithms, and interfaces at a more granular level.

Database Design: If the system involves a database, design the database schema, relationships, and data storage mechanisms. Consider data integrity, normalization, and performance optimization.

User Interface (UI) Design: Create the user interface design, including layout, navigation, and visual elements. The goal is to ensure a user-friendly and intuitive experience based on the requirements gathered earlier.

System Interface Design: Specify the interfaces between various system components, both internal and external. This includes defining how different modules will communicate and exchange data.

Security Design: Incorporate security features into the design, addressing issues such as authentication, authorization, data encryption, and protection against common security threats.

Error Handling and Recovery Design: Define how the system will handle errors, exceptions, and unexpected situations. Develop strategies for error detection, reporting, and recovery to ensure robustness.

Performance Design: Consider the performance requirements outlined in the earlier phases and design the system to meet those requirements. This may involve optimizing algorithms, data structures, and other aspects for effciency.

Scalability and Extensibility: Design the system to be scalable, accommodating potential growth in user base or data volume. Also, consider extensibility to facilitate future updates and additions to the system.

Prototyping: Create prototypes or mockups to validate the design and gather feedback from stakeholders. Prototyping can help identify design flaws and usability issues early in the process.

Documentation: Document the design decisions, specifications, and rationale. This documentation serves as a guide for developers during the implementation phase and as a reference for future maintenance.

Review and Validation: Conduct design reviews to ensure that the proposed design meets the requirements and is feasible for implementation. This step helps catch potential issues before they become more costly to address.

4.Development:

Coding: Write the source code according to the specifications and design documents. Developers follow coding standards and best practices to ensure code quality, readability, and maintainability.

Programming Languages: Choose and use appropriate programming languages for the development based on the project's requirements and technical specifications.

Version Control: Use version control systems (e.g., Git) to manage and track changes to the source code. This helps in collaboration, code review, and the ability to roll back to previous versions if needed.

Code Reviews: Conduct code reviews to ensure that the code meets quality standards, follows best practices, and aligns with the design specifications. This is a collaborative process involving multiple developers.

Unit Testing: Developers write and execute unit tests to ensure that individual units or components of the code work as intended. This helps catch and address bugs at an early stage.

Integration Testing: Combine individual units or modules and perform integration testing to ensure that different parts of the system work together seamlessly. This is crucial for identifying issues that may arise when components are integrated.

Debugging: Identify and fix bugs and issues that arise during the development process. Debugging is an ongoing activity to ensure the correctness of the code .

Documentation: Document the code to provide insights into its functionality, usage, and any special considerations for future developers or maintainers. This documentation is an essential part of knowledge transfer.

Refactoring: Refactor code as needed to improve its structure, efficiency, or maintainability. This might involve reorganizing code, optimizing algorithms, or improving variable names.

Performance Tuning: Optimize the performance of the code by identifying and addressing bottlenecks or inefficiencies. This may involve profiling the code and making improvements based on the findings

Security Implementation: Implement security measures according to the design specifications, such as encryption, secure authentication, and input validation.

Collaboration: Foster collaboration among team members through regular communication and collaboration tools. This is crucial for addressing challenges, sharing knowledge, and maintaining a cohesive development effort.

Adherence to Coding Standards: Ensure that the code follows established coding standards and guidelines. Consistency in coding practices makes the codebase more manageable and readable.

Continuous Integration (CI) and Continuous Deployment (CD): Implement CI/CD pipelines to automate the process of integrating code changes and deploying the application. This helps in maintaining a reliable and consistent development and release workflow.

5.Testing:

Types of Testing:

Unit Testing: Individual units or components are tested in isolation to ensure they work as intended.

Integration Testing: The interaction between different components or modules is tested to identify issues that may arise when they are integrated.

System Testing: The ensure system is tested as a whole to verify that it meets the specified requirements.

Acceptance Testing: The software is tested to ensure that it satisfies user or customer requirements and is ready for deployment.

Manual Testing: Testers manually execute test cases without the use of automated tesng tools. This type of testing is often employed for exploratory testing and user interface testing.

Automated Testing: Use automated testing tools to streamline the testing process, especially for repetitive and time-consuming tasks. Automated testing is efficient for regression testing and ensuring the stability of the code base.

Test Planning: Develop a comprehensive test plan that outlines the testing strategy, scope, resources, schedule, and deliverables. The test plan serves as a guide for the testing process.

Test Cases: Create test cases based on requirements and design specifications. Test cases define the conditions and steps to be followed during testing to verify that the software behaves as expected.

Execution of Test Cases: Execute the test cases against the software to identify defects or deviations from expected behavior. Record the results of each test.

Regression Testing: Perform regression testing to ensure that new code changes do not negatively impact existing functionality. This is crucial to catch unintended side effects.

Performance Testing: Assess the performance of the software under different conditions, such as varying loads and usage patterns. This includes load testing, stress testing, and scalability testing.

Security Testing: Evaluate the security features of the software to identify vulnerabilities and ensure that sensitive data is protected. This includes penetration testing and code analysis for security issues.

User Acceptance Testing (UAT): Allow end-users or stakeholders to test the software in a real-world environment to ensure that it meets their expectations and requirements.

Defect Tracking: Log and track defects or issues identified during testing. Prioritize and address these issues to improve the quality of the software.

Documentation: Document the testing process, including test plans, test cases, test results, and any issues identified. This documentation provides insights into the testing effort and helps with future maintenance.

Sign-off and Release: Obtain approval from stakeholders or the quality assurance team to signify that testing is complete and the software is ready for release.

6.Deployment:

Release Planning: Plan the release of the software, considering factors such as the deployment schedule, potential downtime, and communication strategies for informing stakeholders about the release.

Environment Setup: Prepare the production environment to ensure that it is configured appropriately for the software to run smoothly. This includes seeing up servers, databases, and any necessary infrastructure.

Data Migration: If applicable, migrate data from the testing or development databases to the production database. This ensures that the production environment has the most up-to-date information.

Configuration Management: Ensure that the configuration settings for the production environment match those used during testing and development. This includes database connections, API endpoints, and other configuration parameters.

Deployment Automation: Use deployment automation tools and scripts to streamline the deployment process. Automation reduces the risk of errors and ensures consistency across different environments.

Monitoring and Logging: Implement monitoring and logging mechanisms to track the performance of the software in the production environment. This helps identify issues and ensures a proactive response to potential problems.

User Training: If significant changes have been made to the user interface or workflow, provide training sessions for end-users to familiarize them with the new features and functionalities.

Communication: Communicate with stakeholders, including end-users, about the deployment schedule, expected downtime, and any changes or improvements introduced in the new release.

Post-Deployment Testing: Conduct post-deployment testing to ensure that the software functions as expected in the production environment. This may include additional testing to verify performance, security, and overall system stability.

feedback Collection: Gather feedback from end-users and stakeholders after deployment. This feedback is valuable for identifying any unforeseen issues and for making further improvements in future releases.

Documentation Update: Update documentation, including user manuals and support documents, to reflect any changes introduced in the new release. This helps users and support teams understand the updated features and functionalities.

Handover to Operations and Support: Hand over responsibility for ongoing operations and support to the relevant teams. This includes providing documentation, training, and any necessary information for maintaining the deployed software.