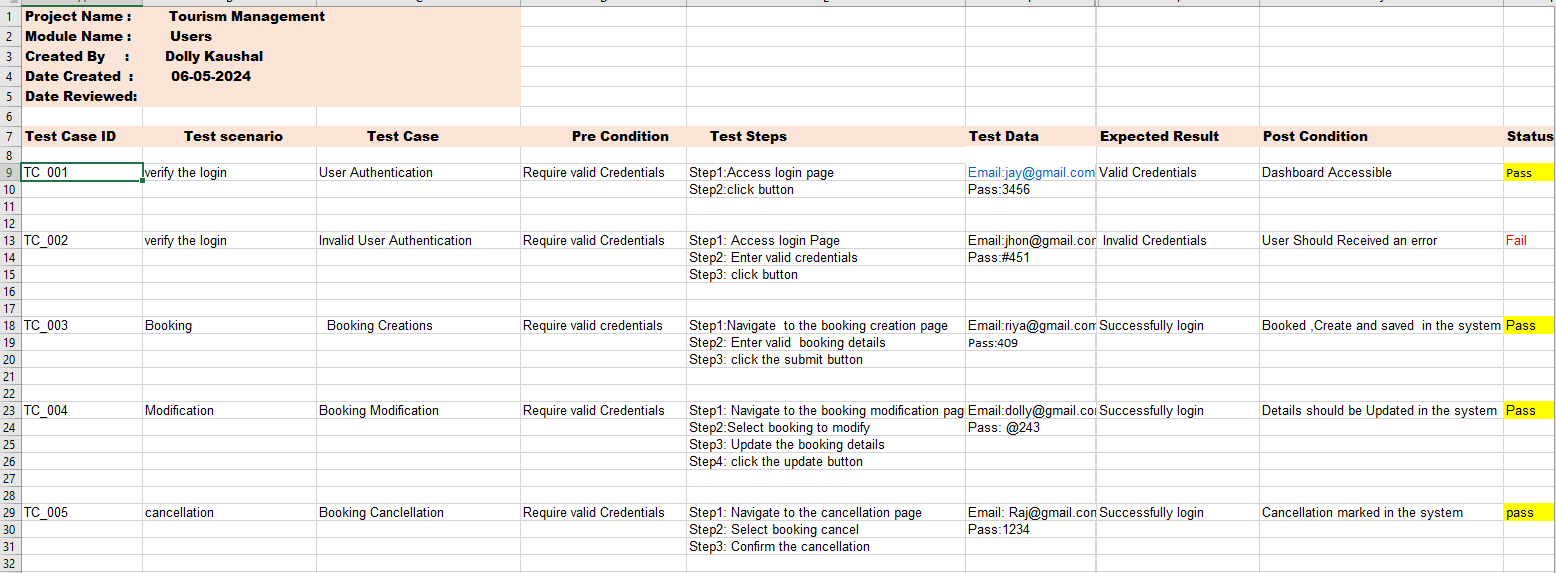
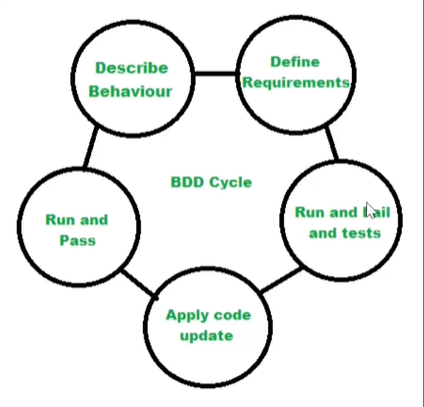
Assignment-1

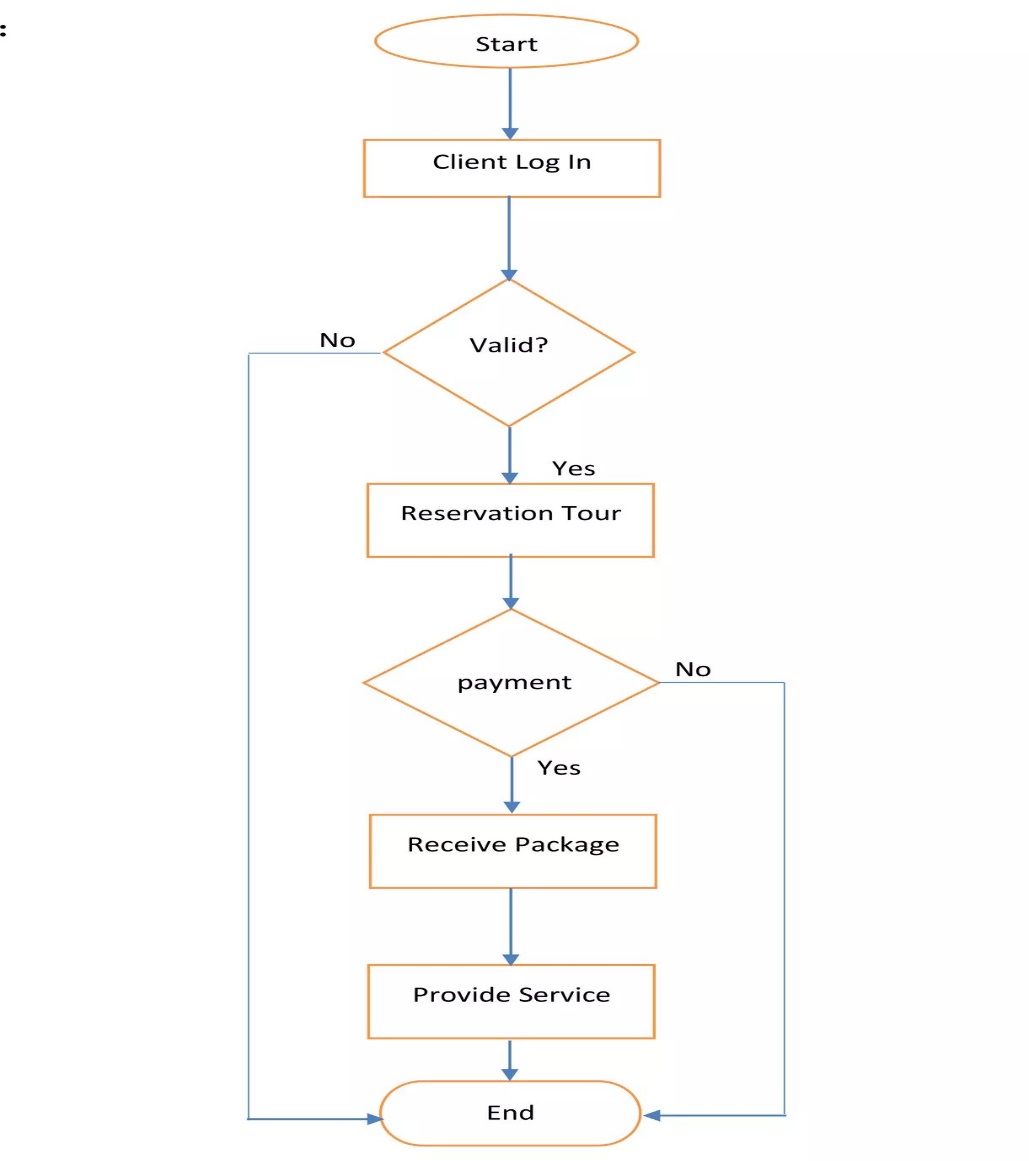
Question-1 Create an infographic illustrating the Test -Driven Development(TDD) process. Highlight steps like writing tests before code, benfits such as bug reduction and how it fosters software reliability?

Answer:

TDD(Test Driven Data):



Flowchart:



Test-Driven Development (TDD) offers several benefits, including bug reduction and fostering software reliability:

Early Bug Detection: TDD requires writing tests before writing the actual code. This approach helps in identifying potential bugs or issues at an early stage of development. Since tests are written based on expected behavior, any deviation from the expected behavior indicates a bug. By addressing these issues early in the development cycle, the overall number of bugs in the software is significantly reduced.

Improved Code Quality: TDD encourages developers to write modular and testable code. Writing tests first forces developers to think about the design and functionality of their code from the outset. This leads to cleaner, more maintainable code that is easier to understand and modify. As a result, the overall code quality improves, reducing the likelihood of introducing bugs during development.

Regression Testing: TDD promotes the practice of running automated tests frequently. As new features are added or existing code is modified, developers can quickly run the tests to ensure that the changes haven't introduced any unintended side effects or regressions. This continuous validation helps in maintaining the reliability of the software over time, as it prevents previously fixed bugs from resurfacing due to code changes.

Refactoring Safety Net: TDD provides a safety net for refactoring existing code. When refactoring code to improve its design or performance, developers can rely on the existing tests to ensure that the behavior of the software remains unchanged. Since TDD encourages a comprehensive suite of tests, developers can refactor with confidence, knowing that any regressions will be caught by the existing tests.

Improved Collaboration: TDD encourages collaboration between developers and stakeholders by providing a common language for discussing requirements and expectations. Since tests serve as executable specifications of the system's behavior, they facilitate communication between team members, including developers, testers, and product owners. This shared understanding leads to better alignment of expectations and ultimately results in a more reliable software product.

Reduced Debugging Time: By detecting bugs early and providing a suite of automated tests, TDD reduces the time spent on debugging and troubleshooting issues. Since bugs are identified at the unit level, they are typically easier to isolate and fix. Additionally, the automated nature of the tests means that they can be run quickly and repeatedly, allowing developers to pinpoint and resolve issues more efficiently.

Question-2 Produce a comparative infographic of TDD, BDD, FDD methodologies. Illustrate there unique approaches, benefits , suitability for different software development contexts.Use visual to enhance?

Answer:

|  |  |  |  |
| --- | --- | --- | --- |
|  | TDD | BDD | FDD |
| Approaches | 1.Write a failing test  2. Write the minimum code to make the test pass.  3. Refactor code while keeping the tests passing. | 1. Describes the initial context or preconditions.  2. Describes the action or event that triggers the behavior.  3. Describes the expected outcome or behavior. | 1.Feature List  2. Development by Feature |
| Benifits | 1. Bug Reduction  2. Improved Software Reliability. | 1. Enhanced Collaboration.  2. Focus on Business Value. | 1.Incremental Delivery  2.Clear Accountability |
| Suitability | 1.Small to Medium project.  2. Agile and Iterative Development | 1. User-Centric Design  2. Agile Development  3. Complex Systems | 1. Large-Scale Projects.  2.Client-Focused Projects. |