```
+--------------------------------------------------+
|            C# Development System                 |
+--------------------------------------------------+
|             System Specification                 |
+--------------------------------------------------+
|            Implementation Cases                  |
|--------------------------------------------------|
| 1. Implement User Validation                     |
| 2. Create Dependency Injection Service           |
| 3. Develop Factory Method                        |
| 4. Establish Observation Mechanism               |
| 5. Implement Singleton Manager                   |
+--------------------------------------------------+
```

Descriptions of Implementation Cases:

1. **Implement User Validation:**
   - **Description:** Develop and test a user validation component to ensure that the input data provided by users is valid and secure.
2. **Create Dependency Injection Service:**
   - **Description:** Implement a dependency injection service to allow system components to be easily injected and swapped, promoting loose coupling and testability.
3. **Develop Factory Method:**
   - **Description:** Create a factory method that encapsulates object creation logic, allowing flexibility in creating different types of objects as needed.
4. **Establish Observation Mechanism:**
   - **Description:** Develop an observation mechanism that allows observer objects to be automatically notified of changes in the

state of an observed object, promoting asynchronous communication between system components.

5. **Implement Singleton Manager:**
   - **Description:** Create a singleton manager that ensures only one instance of a particular class is created and provides a global access point to that instance throughout the system.

```
+-------------------------------------------------+
|        Implementation of Test Coverage        |
|              in C# Project               |
+-------------------------------------------------+
|              Use Case                  |
+-------------------------------------------------+
|   Use Case: Implement Test Coverage          |
+-------------------------------------------------+
|          Primary Actor                  |
+-------------------------------------------------+
|        Software Developer                |
+-------------------------------------------------+
|            Preconditions                 |
+-------------------------------------------------+
| The source code of the C# project is available   |
| for modification and testing.                |
+-------------------------------------------------+
|           Post-Conditions                |
+-------------------------------------------------+
| Adequate test coverage is implemented in the    |
| C# project to ensure software quality and        |
| reliability.                          |
+-------------------------------------------------+
|          Main Flow                     |
+-------------------------------------------------+
| 1. The developer identifies critical areas of    |
|   the code that need test coverage.           |
|                                  |
| 2. The developer writes unit tests for each of    |
|   the identified functionalities or methods.    |
|                                  |
| 3. The tests are locally executed to ensure they |
|    pass and cover the functionalities as        |
|    expected.                             |
|                                          |
| 4. If the tests fail, the developer identifies   |
|    and corrects issues in the code.           |
|                                          |
| 5. After ensuring all tests pass locally, the    |
|    code and tests are submitted for continuous   |
|    integration (CI) or a build server for      |
|    automatic test execution in a testing       |
|    environment.                          |
+-------------------------------------------------+
|              Extensions                   |
+-------------------------------------------------+
| Extension 1: Implement Integration Tests        |
| - The developer may extend the use case to      |
|   include the implementation of integration     |
|   tests to ensure proper interaction between    |
|   different components of the system.          |
|                                          |
| Extension 2: Update Test Coverage            |
| - The developer may extend the use case to      |
|   include continuous updating of test coverage  |
|   as the code is modified or new functionalities|
|   are added.                             |
+-------------------------------------------------+
```