Functional Documentation- Key Components

Functional documentation is essential for defining the **business logic, workflows, and user interactions** of a system.

Below are its key components:

1. Requirements

- **Business Requirements:** Defines the overall business objectives and needs.
- **Functional Requirements:** Specifies what the system should do (e.g., features, capabilities).
- **Non-Functional Requirements:** Includes performance, security, and usability constraints.

2. Workflows

- **Process Flow Diagrams:** Visual representation of how users interact with the system.
- **State Transitions:** Defines different system states and their conditions.
- **User Roles & Permissions:** Describes how different users access features.

3. Use Cases

- **Actors:** Identifies the users or systems interacting with the application.

- **Steps & Scenarios:** Defines actions users take and expected results.
- **Exception Handling:** Specifies how the system responds to errors or edge cases.

1. Requirements

1.1 Business Requirements

- Define the overall objectives and purpose of the system.
- Example: "The system should allow users to search for books in an online library."

1.2 Functional Requirements

- Describe system features and expected behavior.
- Example:
    - Users can register and log in.
    - Users can borrow and return books.
    - Librarians can add, edit, or remove books.

1.3 Non-Functional Requirements

- Outline performance, security, and usability constraints.
- Example:
    - The system should handle 5000 concurrent users.
    - Data must be encrypted for security.

2. Workflows

2.1 Process Flow Diagrams

- Illustrates how users interact with the system.
- Example: A diagram showing user login, book search, and checkout process.

2.2 State Transitions

- Defines different states of an entity in the system.
- Example: A book can be in "Available," "Checked Out," or "Reserved" states.

2.3 User Roles & Permissions

- Describes access levels for different users.
- Example:
    - **Member**: Can search and borrow books.
    - **Librarian**: Can manage book inventory.
    - **Admin**: Has full access to system settings.

3. Use Cases

3.1 Actors

- Identifies users interacting with the system.
- Example:
    - **Primary Actors**: Member, Librarian, Admin.
    - **External Systems**: Payment Gateway for fines.

3.2 Steps & Scenarios

- Defines interactions and expected outcomes.
- Example: **Use Case: Borrow a Book**
    1. Member searches for a book.
    2. If available, member requests to borrow it.
    3. System updates book status to "Checked Out."

3.3 Exception Handling

- Describes system behavior in error conditions.
- Example:
    - If a book is not available, the system shows "Waitlist Option."
    - If a member exceeds the borrow limit, an error message appears.

This document serves as a structured **template** for functional documentation.