Functional Requirements:

Plant & Greenhouse:

- Plant Creation: The creation of plants by specific type. -Factory Method.
- Plant Lifecycle Management: The system needs to manage a plant lifecycle, where its behaviour changes based on its state. -State
- Flexible Plant care: The system has different plant care strategies dependant on needs that can be easily changed. -Strategy.
- Plant Monitoring: Notifies all relevant staff when a plants needs change or when it's state changes (ready for sale, etc.) – Observer

Inventory & Stock:

- Simplified Inventory Interface: An interface for staff and customers to perform complex inventory operations (check stock, filter, etc.) – Façade.

Staff & Tasks:

- Staff Task Management: The system encapsulates staff actions as objects allowing them to be queued, logged and executed. – Command
- Staff-Client co-ordination: The system manages all communication between customers and staff through a mediator to avoid tangled references. – Mediator
- Escalation of requests: The system passes complex customer requests through a chain of staff members until it is handled. – Chain of Responsibility.

Customer & Sales:

- Purchase Customisation: The system will allow dynamically added decorations to their purchases, modifying the price and descriptions. -Decorator
- Bundle Purchases: The system allows for bundle products such as kits, treating individual and bundle items uniformly. – Composite
- Complex order construction: The system creates complex orders step by step allowing for different variations. – Builder

Non – Functional

- The system is designed to be easily maintainable and easily extendable, allowing new plant types, staff roles, customer actions and payment methods to be added with minimal changes to existing classes and code. This will be achieved by using interface and design patterns like strategy and decorator. -scalability
- The system will ensure data consistency, particularly for inventory management. Critical operations like sales transactions will be updated correctly and customers cannot purchase unavailable stock. – Reliability
- The codebase shall be organized into discrete modules (plant management, staff system, sales) with clear interfaces, allowing teams to work on separate modules simultaneously. – modularity