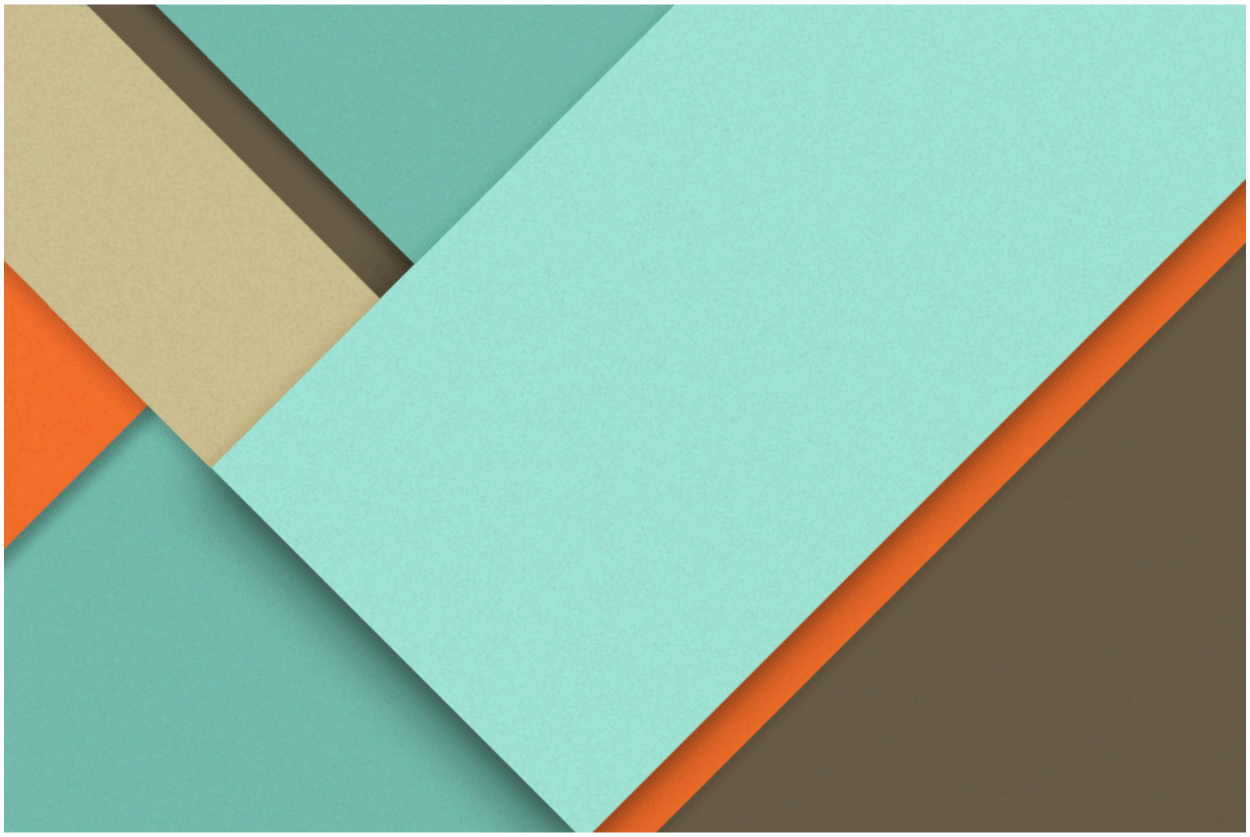

COS214 Project Document

Overview

The Functional Requirements as they currently stand are listed first, followed by a list of the the patterns we discussed, the ones from the At Least 5 list are highlighted in bold. Details for each as given by the teams are discussed in the section below those, please keep in mind there were a lot of sources to work off of and not everything was super clear or discussed in the meeting so there may be duplicates, gaps and misunderstandings.

Functional Requirements

1. **Greenhouse & Inventory**
    a. The system must allow new stock to be added easily
    b. The system must provide an inventory tracking subsystem to catalogue all plants, seeds and stock items
    c. Plants must be represented in different life cycle states (e.g. seedling, growing, mature, ready for sale)
    d. Different plant types must require different care strategies (e.g. watering frequency, sunlight exposure, soil type, pruning)
2. **Staff Responsibilities**
    a. Staff must be able to tend to plants according to their unique care routines (watering, fertilizing, sunlight management, pruning, etc)
    b. Staff must monitor plant health and life cycles, and take action when plants need extra attention
    c. Staff must track and update the inventory to ensure sales floor availability reflects greenhouse stock
    d. Staff must manage the sales floor, assisting customers and maintaining plant displays
    e. Staff must coordinate their tasks with one another to ensure smooth nursery operations
    f. The system must support extension of staff roles, allowing for promotion or acquisition of new skills (e.g. delivery staff, landscapers, greenhouse managers)

3. **Customer Interactions**

    a. Customers must be able to browse and view available plants

    b. Customers must be able to compare plants side by side (e.g. price, care requirements, size, growth rate)

    c. Customers must be able to filter plants by type, price range, or care level

    d. Customers must be able to request information about plants

    e. Customers must be able to request system/staff recommendations

    f. Customers must be able to customize plants and orders (e.g. decorative pots, gift wrapping, special arrangements)

    g. Customers must receive purchase assistance from staff

    h. Customers must be notified of stock availability and updates

    i. Customers must be able to revert to a previous stage when building their order (undo/redo style functionality)

    j. Customers may review their purchase history (?)

4. **Customer & Staff Coordination**

    a. Staff must be able to manage multiple customer requests efficiently

List of Design Patterns

1. **Iterator**
2. Builder
3. **Observer**
4. **Mediator**
5. **Command**
6. **Facade**
7. **Strategy**
8. **Decorator**
9. Memento
10. **Abstract Factory**
11. **Composite**
12. **State**
13. **Chain of Responsibility**
14. Template
15. Adapter (?)

Avoid Singletons

Specifications

Staff Plant Care Execution (Design Pattern: Strategy & Command)

The system must allow staff to execute unique care routines on plants, adapting to each plant's specific needs (watering, fertilizing, pruning, sunlight management). Strategy enables dynamic selection of care algorithms without hardcoding behaviors, while Command encapsulates actions as reusable tasks, supporting flexibility and potential undo/redo.

Plant Monitoring (Design Patterns: Observer & State)

Plants must notify staff when they require attention (e.g. watering, fertilizing, reaching maturity or declining health). Observer ensures plants act as subjects that broadcast events, while staff act as observers who react. The State pattern models plant life cycles (seed, growing, sellable, dying), allowing transitions to occur seamlessly and triggering notifications.

Staff–Inventory Coordination (Design Pattern: Mediator & Composite)

The system must maintain a real-time, centralised inventory that reflects both greenhouse and sales floor stock. Staff must coordinate updates without direct dependencies. A Mediator could act as the hub for communication, routing updates between staff and inventory. Composite structures the inventory into collections of plants in different states, ensuring plants move between categories (e.g. from greenhouse to sales floor) smoothly.

Coordination of Customer-Staff Interactions (Design Patterns: Mediator and Command)

The system has to facilitate interactions between customers and staff, such as routing requests for guidance, recommendations or purchase assistance, while handling sales transactions that update inventory and record details. Mediator could act like a central hub to manage communication between objects (e.g. customers, staff, inventory system, etc) and Command could be useful for transactions, where a purchase command can update inventory, record details and handle rollbacks if needed.

Staff–Customer Interaction Handling (Design Pattern: Chain of Responsibility & Mediator)

Customer queries should be handled by staff in sequence or escalated to managers if needed. Chain of Responsibility passes requests along staff roles, while Mediator centralises routing to avoid tightly coupled interactions.

Adding of New Stock (Design Pattern: Factory)

The system must allow new stock to be introduced easily. Abstract Factory supports the creation of families of plant types without tightly coupling code to specific classes.

Plant States (Design Pattern: State)

State captures transitions between stages (seed/growing/sellable/dying).

Customer Browsing and Viewing Available Plants (Design Pattern: Iterator)

Customers must be able to browse and traverse lists of available plants.Iterator allows iteration without exposing or modifying the underlying collection structure.

Personalisation of Plant Purchases (Design Patterns: Decorator)

Customers must be able to customize plants (e.g. decorative pots, gift wrapping, special arrangements). Decorator dynamically adds optional features without modifying the base plant object.

Real-Time Updates on Stock Changes and Availability (Design Pattern: Observer)

The system must notify staff and customers about stock changes (restocks, low stock, sold out). Observer ensures real-time propagation of updates between greenhouse, inventory, staff and sales floor.

Purchase Process Simplification (Design Patterns: Facade & Bridge)

The system must allow customers to purchase plants through a simplified interface that hides underlying complexity (inventory checks, staff coordination, payment handling). Facade provides a unified interface, reducing coupling and ensuring a smoother experience. Bridge for client or staff.

Order Undo/Redo and Customisation History (Design Pattern: Memento)

Customers must be able to revert their plant customisation or return to a previous state during order building. Memento captures snapshots of the order state, enabling safe undo/redo without exposing internal object details.