Guilded_Rose_WM
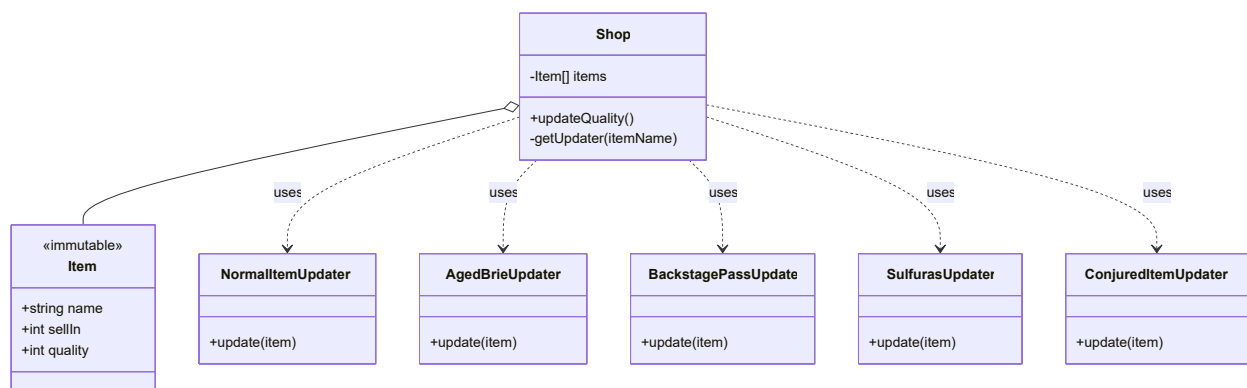
Due Date (1)	Type	Status
January 30, 2026	link	complete

Work

Your job will be to propose, defend, and implement a new design for this code.

- ✅ Evaluate the code (**Gilded_Rose.js**) , potential improvements. ✅ 2026-01-30
- ✅ Create a class diagram. ✅ 2026-01-30
- ✅ Write up a small description of the major issues ✅ 2026-01-30
    - ✅ Explain the design choices ✅ 2026-01-30
    - ✅ Discuss how they contribute to the solving of those problems. ✅ 2026-01-30
    - ✅ Potential drawbacks ✅ 2026-01-30

UML Diagram

Notes

Issues:

- The code suffers from deeply nested conditionals with multiple hardcoded string comparisons which forms tight coupling between the Shop class and specific item names. Adding a new item type

requires modifying the `updateQuality` method, risking unintended side effects on existing items.

Design Choices:

- I applied the Strategy pattern by creating dedicated updater classes for each item type.
- The Shop delegates update logic to the appropriate strategy based on item name.
  Benefits:
  1. Decoupling - Shop no longer contains item-specific logic; each updater encapsulates one item's behaviour
  2. Extensibility: New items require adding a new updater class, not modifying existing code
  3. Complexity: Eliminates nested ifs—each updater has simple, linear logic